CHAPTER

9

DESIGNING FOR INTERNET AND INTRANET SERVICES

After reading this chapter and completing the exercises, you will be able to:

- Identify common design considerations for providing an infrastructure for services to the Internet
- Understand the steps in designing an Internet site infrastructure
- Identify common design considerations for providing an infrastructure for services to an intranet
- Understand the steps for providing an intranet site infrastructure

In the last chapter, we focused on providing Internet access to your internal users so that they can reach WWW, FTP, SMTP, and other services that other organizations have made available. In this chapter, we reverse the roles and take a look at providing those same services to the rest of the world through the Internet. We then look at providing those services to internal customers through an intranet. Finally, we discuss some special considerations that can complicate your life on the Internet if you're not careful.

What Your Design Needs When Providing Services to the Internet

There are estimates about the number of people using the Internet, but nobody really knows how many there are. It is certainly in the millions worldwide and, just as important, nobody knows who they are or where they're coming from. So, for your design to be successful, it must have extreme scalability.

Your site also must be secure. There are six billion people on the planet, and they include customers, partners, competitors, hackers, crackers, and mischievous 12-year-olds. You can bet that once you've connected your servers to the Internet, some of these folks will not behave themselves. So to avoid a visit from your CIO after he reads his name in the morning paper next to "Web site defaced and 100,000 credit card numbers stolen," you must incorporate security into your Internet site design.

Your site also must be available. Your users probably work from 7 a.m. to 6 p.m. or so, and many Internet sites are heavily used 24/7. This might not seem like an important distinction, because you don't turn off all your servers, routers, and switches when you lock the office doors and go home each night. But if users are on your site all day long, when can you upgrade hardware and software? What about routine maintenance tasks, such as reindexing databases? Even content updates can be tricky. Fortunately, Microsoft and others have provided several useful tools that make it possible, if not simple, to design and implement Internet sites that accommodate most needs.

Providing an Infrastructure for Services to the Internet

Before we jump into designing our network, we need to discuss some design considerations. From previous chapters, you should have an appreciation of the factors that drive and shape a design. Although these factors are fairly common throughout networking, they vary by application; so we discuss them in detail below. We then show you an infrastructure design that implements some of these considerations.

Design Considerations by Type

Design requirements are critical to a successful project. In this section, we look at design considerations that are important for delivery of services to the Internet. We discuss why corporate Internet sites are popularly divided into two classes—B2B and B2C. In addition, we discuss some design considerations for the following generic types of services: e-mail, file access, and special media.

Common B2B Design Considerations

B2B—or business to business—describes a corporate Internet presence that allows companies to make business transactions or share information with each other. Loosely

translated, B2B is the next generation of **Electronic Data Interchange (EDI)** with a WWW user interface. EDI is a standard for the exchange of electronic data between businesses.

Although every site is different, most B2B sites have these characteristics in common:

- Low number of users, usually in the hundreds
- High security needs due to proprietary information or financial transactions
- High reliability because downtime can easily cost millions of dollars per hour
- Low availability because business hours are usually restricted and transactions are frequently on a schedule that can be weekly or monthly

The first thing to do when designing a B2B solution is understand the data you'll be transporting, where it comes from, and where it goes. Frequently, the source or destination of your data will be a mainframe or some other large corporate information system. Because these systems aren't exactly portable or cheap, you will often have to design around them.

Most mainframes are capable of running rather impressive Web services these days, but you certainly won't be able to toss your mainframe and connected systems into a rack at your favorite ISP's hosting facility. Because this is a Microsoft book, we assume you'll be connecting your data center to IIS Web servers. This can be a complex task for several reasons, and you'll probably be thankful that you read all about SNA and UNIX connectivity in Chapter 5.

The complexity of the integration is largely a result of the lack of similarity between the IBM OS/390 world and the PC world at every layer of the OSI model. For instance, at the Physical and Data Link layers (layers 1 and 2), you'll want to use CAT 5 and Ethernet while the mainframe will probably be using **Escon**. At the Network layer (layer 3), you'll have to use TCP/IP for Internet connectivity, but the mainframe will probably be using **Systems Network Architecture (SNA)**. At the Presentation layer (layer 6), you may want to receive data in ASCII text files, but the mainframe will want to send it in **Extended Binary-Coded Decimal Interchange Code (EBCDIC)**.

You may prefer to receive files through FTP, but unless your mainframe is up-to-date, you could be stuck with less-than-desirable downloads. Fortunately, things aren't quite so difficult with UNIX, because UNIX systems have been natively using TCP/IP since the beginning. Nevertheless, they usually employ the same **batch-oriented processing systems** as mainframes, where the PC world is often **event-driven**. This latter issue is typically a developer's problem, not MCSE's, but it's still good to understand because you will have to support the development staff.

The important thing to realize here is that you should not take physical or logical connectivity to legacy systems for granted. It may not be as easy as it looks and you should always perform a **proof-of-concept** test before sinking a lot of money into a design.

Make sure you do a thorough discovery process and allow plenty of time in your project plan to research alternate solutions and create a backup plan.

You can test your plan with a proof-of-concept test, a scaled-down version of an entire project. It uses all the components, so interoperability can be assured without the cost of the entire project. This is an excellent way to mitigate risk because you have very little invested in a solution.

B2B Security

Confidentiality is preventing unauthorized access to protected data. Confidentiality obviously implies that some people are allowed to see the data and others aren't. That means you need to have two things: a way to distinguish between those two groups of people and a way to hide the data from the latter group.

There are two common ways to distinguish between these groups. The first is user accounts, with which you should already be familiar. In a typical B2B scenario, not only will there be a relatively small number of users, but each of them will typically have signed a contract with you for your services. This means that managing user accounts shouldn't be too difficult, especially if you're using Active Directory.

The second popular way to distinguish between these groups of people is by using digital certificates or security tokens. **Digital certificates** are small files that reside on a user's computer. These certificates are verified by querying a public registry such as Verisign. Prepare to pay some bucks for this service. If you're looking for something a little cheaper (read: free), you can use the Certificate Server that ships with Windows 2000 Server products. These use the ITU-T recommendation X.509 standard for certificates as part of a directory structure. This is a requirement to establish a Secure Sockets Layer (SSL) connection.

Security tokens are small pieces of hardware that contain a tiny microchip that displays a number that changes every minute and is synchronized with an application that runs on your server. In order to access a system, this number must be entered when the user logs in. Because it changes every minute, it is much more secure than static passwords. Your executives can write this number on Post-It notes and stick them to their laptops all they want now; 60 seconds later, you're secure again! Obviously, most e-commerce sites would love to have this level of security, but the cost of the hardware and administering security tokens for several million customers is out of the question. However, it is feasible for a typical B2B site. For more information about tokens, check out some of the more popular vendor sites, such as RSA at *www.rsasecurity.com/*.

Digital signatures and security tokens allow you to be fairly confident that the person on the other end of the wire is who they say they are. When you combine one or both of these tools with normal user names and passwords, it's often referred to as **strong authentication**. For more information on digital signatures, start with the Web site of the National Institute of Standards and Technology (NIST), *www.itl.nist.gov/fipspubs/*,

which published Federal Information Processing Standard (FIPS) PUB 186, also known as the Digital Signature Standard (DSS). This was originally proposed in 1991 and revised in 1993, 1996, and 2000. Don't expect to see too much of this on the exam, but you'll want to understand it before you design a secure site.

Unless you're in the CIA, you're going to be using conventional encryption such as **Data Encryption Scheme (DES)**, **Triple DES**, or **Blowfish** to hide your data. If you are in the CIA, you can avail yourself of many other techniques such as **steganography**, which attempts to conceal the fact that there is a message (typically by hiding it in another message), but if you're in the CIA, you probably already know this. For the rest of us, Microsoft has provided tools in nearly every product to encrypt data, from SQL Server to Outlook Express. A quick search of Microsoft's knowledge base will turn up more than you ever wanted to know; so, in the following pages, we'll just touch on the facts that are critical to your design success.

We start by making sure you understand that there are two kinds of encryption schemes: unconditionally secure and computationally secure. If an encryption scheme is **unconditionally secure**, it means that it is impossible to break, simply because the necessary information is not included in the message, no matter how much time or resources you have. With the exception of one-time pads, there aren't any encryption schemes in use that are considered unconditionally secure. The rest of the schemes rely on being **computationally secure**, which means that you can break the encryption with **brute force** by trying every possible key, but that process would take so long that by the time you're finished, the data is no longer important, or it costs more to crack than the data is worth.

For more information on one-time pads, visit the FAQ section of *http://web.ranum.com/pubs/otpfaq/*.

The standard versions of Microsoft products that include 56-bit encryption and free downloads are available to support 128-bit Triple DES. So, you might be saying to yourself, "No problem! I'll just spend as much money as I need to encrypt." Well, the problem is that while users inside the United States are free to use the commonly available 128-bit algorithms (such as Triple DES), law prohibits setting up an encrypted link or sending an encrypted message overseas using more than 56 bits. It goes without saying that in today's world of global e-business, this is a major problem. But the debate in the legislature continues and with rapid advances in technology, encryption laws are sure to change; so make sure you verify that your design doesn't break the laws of any country in which you do business.

As you consider the choices of encryption, remember that the stronger the encryption, the more processing power it takes. The process of encrypting and decrypting is extremely CPU-intensive. With today's super-fast processors, this isn't too much of a concern for typical user traffic. However, encrypting a lot of data and especially encrypting a large number of simultaneous transactions can easily drag almost any server to its knees. In this instance, you could evaluate one of the special expansion boards designed

to offload the encryption from your CPU to dedicated processors or ASICs (Application-Specific Integrated Circuits). In fact, actual encryption can take place at many points along the network. Remember that in practical terms, encryption creates a tunnel through which devices on each end can "see," but an observer in the middle cannot "see" the data inside the tunnel. As a designer, it is important to realize your options.

Microsoft has provided encryption tools, such as the aptly named Point-to-Point Tunneling Protocol (PPTP), which can be terminated on a PC in a home or remote office so that every packet leaving the NIC is encrypted. If the other end is an office with many servers and other devices to talk to, you could actually set up a separate tunnel from the PC to each device, but that would be a maintenance nightmare. A much better solution is to terminate this tunnel in a router or dedicated VPN concentrator appliance, such as Nortel's Contivity or Cisco's Altiga. On most B2B sites, you should be OK, but if you plan on running several processor-intensive applications, such as SQL Server or Exchange, on the same server, then keep this in mind.

Another fact that you should be aware of is that current encryption standards don't play well with many other technologies. IPSec is famous for its incompatibility with NAT, although workarounds are being developed. Others may not be completely compatible with Active Directory or other Microsoft products, and security protocols in general definitely don't like being load balanced. Again, thorough testing and a proof-of-concept procedure should be performed before employing new technology.

B2B Integrity, Nonrepudiation, and Reliability

Integrity refers to whether the data is complete, sound, and unimpaired. As it relates to network security, for example, this can be satisfied by the use of IPSec with Authentication Header (AH), which would specify SHA or MD5 as a data integrity checksum. This checksum works just like checksums at the Data Link and Network layers, except that it's built into the packet to prevent tampering. Most other encryption schemes have some sort of mechanism to ensure integrity.

Nonrepudiation means that once someone sends you a message (a transaction, for instance), you can prove they sent it. Conversely, once you receive a transaction, the other party can prove you received it. This is the electronic equivalent of parcel tracking numbers used by FedEx and UPS. Once someone gives you the tracking number, you know the package has been picked up and is now in the possession of the courier. In addition, once you sign for a package, the same tracking number can be used to show your signature to the shipper. This is also built into most security algorithms.

For more information on security, check out the International Information Systems Security Certifications Consortium, Inc. (ISC^2) at *www.isc2.org* and the Intiss links page at *www.intiss.com/intisslinks.html*.

Our next discussion point is reliability. Design requirements for reliability vary widely by application. Downtime for some systems obviously is more expensive than others. Whatever your requirements, there are a number of guidelines to follow for reliability. The first is analogous to the weakest link in a chain. Remember that it takes a lot of components working together to make a network operate, and each one of those is a potential point of failure. If your application is important enough to provide fault tolerance for one component, it's usually important enough for most, if not all, of them. While some solutions are considerably cheaper than others (such as using extra hard drives and Microsoft's RAID 5), making others redundant (such as firewalls and load balancers) can cost big bucks. A little redundancy is better than no redundancy, but don't be fooled into thinking a system is bulletproof unless you've tested it by breaking every element and verifying that the system recovers or compensates automatically.

If you are on a restrictive budget, identify your most common points of failure and the points of failure that will take the longest to repair. For instance, hard drives (moving parts) and RAM (extremely sensitive) may fail far more often than CPUs or chassis—so have some spares readily available. In addition, if you lose a power supply, you can usually purchase one at a computer store, but if a backhoe cuts your local loop, you could be waiting days for your telco to fix the problem. Figure out a backup plan that is appropriate for your reliability requirements and budget.

Another money-saving option appears after you come to appreciate the difference between hot spares and cold spares. **Hot spares** are devices that offer Physical layer redundancy for another device, are always powered on, and are capable of detecting a failure and immediately assuming an active role. By contrast, **cold spares** typically sit on a nearby shelf in the original packaging from the manufacturer and must be manually placed into service in the event of an outage. Usually, hot spares are much more expensive than cold spares. For instance, redundant, dedicated T-1 links to the Internet cost more than a single T-1 and an ISDN dial backup link. They also take more time to configure and test—so don't forget to include that in your project plan.

Many people call all spares hot spares because they like saying hot spares, but that doesn't make them so.

Here's a word of warning: Don't forget that all the redundant hardware in the world won't protect you from an application or OS glitch. That's why Microsoft developed Cluster Services. For a site with largely static content, you can have redundant servers with separate copies of the content; if one goes down, the other can take over with no interruption of service.

Unfortunately, the applications commonly found in B2B sites are more complicated than simple page servers. Thus, replication is difficult to achieve. In addition, in Microsoft's architecture, servers are typically the sole owners of a particular database. For instance, your mailbox would only exist on one Exchange server. Thus, if that server goes down,

you would not be able to access it through another Exchange server. Fortunately, with Cluster Services, the second server can be configured to assume ownership of the first server's data, and they may continually update each other with information so that if one goes down in the middle of a transaction, you don't have to start the transaction over from the beginning, since the second server already knows where you left off. The point here is that there are many different levels in which a system can fail. Make sure that your design requirements are explicit regarding reliability.

Common B2C Design Considerations

A **B2C**—or business to consumer—site is an e-commerce or informational site used by consumers. Like B2B sites, every B2C site is different, but most have these aspects in common:

- Support for an extremely large user base
- High security requirement, but different focus
- High availability requirement
- High reliability requirement

As we mentioned in the B2B section, tracking a large number of users can be quite challenging. Because most authentication systems, such as Windows NT's SAM database, cannot scale to millions of user accounts, most B2C sites that need to track users employ one of two methods: cookies or a simple SQL Server database with a user name and password field.

A **cookie** is a file that is placed by the browser onto a user's hard drive. The cookie method is typically transparent to the user and is most effective when users need to be tracked but security isn't an issue. The homemade SQL Server login, however, requires the user to remember and type in their user name and password. This system can be made reasonably secure.

Unlike B2B, where security focused on confidentiality, integrity, and nonrepudiation, the goal of security on B2C sites is often focused on availability, in the sense of preventing **Denial of Service (DoS)** attacks, and integrity, with respect to the system and Web application files on the servers and the packets as they traverse the network.

Protecting against DoS is challenging because there are so many components in the network that can be abused. For instance, a diligent administrator can usually keep intruders from mucking around Web servers, but preventing them from filling up bandwidth with spoofed packets is nigh impossible. While there are some schemes out there that can help you, if they're too expensive or inappropriate for other reasons, at least remember to keep good logs so your forensics team can track the culprits down.

Common Design Considerations for E-Mail

E-mail used to be a fairly simple exercise. You had an SMTP and POP3 services, and DNS and the client software took care of everything else. These days, it's a little more complicated. Users aren't satisfied with ASCII text anymore. They want different fonts, colors, file attachments, and HTML in their e-mail. In addition, don't forget contacts, calendars, and a host of other services that users believe should be integrated into their e-mail solution.

Fortunately, Microsoft helps the situation with their IMAP protocol and Exchange servers, which are popping up around the Internet now to support e-mail. However, as e-mail becomes more widespread, it is becoming more mission critical. So, to keep your job and your sanity, keep the following in mind when designing a site to support e-mail:

- The number of accounts you want to support
- The messaging protocol you want to support and the types of clients
- The size and types of attachments you will allow
- The types of messages you want to send
- Security requirements
- Availability requirements

When you know the facts around your e-mail system, you are better prepared to select specific solutions. For instance, you'll decide whether to choose Active Directory or POP3 accounts, whether to cluster the Exchange service, and whether to run third-party e-mail virus scanners. Whatever you do, make sure you keep up-to-date on the latest patches, and have a plan in place to deal with the next e-mail virus.

Another consideration when creating e-mail services is the type of messages you want to send. Often, the focus of Internet e-mail applications is sending bulk e-mail, whether solicited or unsolicited. There are many programs that do nothing more than pull names and e-mail addresses from a database and send slightly customized form letters. However, these programs typically rely on external SMTP servers, which can be load balanced if necessary.

If you send bulk e-mail, be aware that you may have some dubious company: **spam** (unsolicited or junk e-mail). Many ISPs use so-called **blackhole services** such as mail-abuse.org, which compiles lists of servers known to originate spam. This list is then used to intentionally sever communications with these servers. This means that if you fail to secure your SMTP service, and the evil spammers use your server as a relay to send unsolicited bulk mail, don't be surprised if you start getting complaints from users about their e-mail being rejected. Check the lists for your domain name and then follow the instructions on their Web sites to have yourself removed and restore e-mail connectivity. On the good side, if you can prove you are legitimately sending solicited bulk e-mail, you can contact the major ISPs and be put on their white hat list so that they don't block your e-mail.

Common Design Considerations for File Access

Unlike e-mail, file access on the Internet is much simpler than it used to be. From programs like Kermit, we upgraded to FTP, which has gradually been evolving as it integrates with HTTP, so that command line interfaces are no longer required and you can simply click a link in your Web browser to download a file. In fact, you can transfer files with the HTTP protocol without using FTP at all.

There are still design considerations, of course. For instance, if you're only serving up files and not receiving them (or can implement a separate process for receiving), file access is a good candidate for Microsoft's Network Load Balancing (NLB). A solid, flexible FTP server is included for free in IIS which is, in turn, included for free in Windows NT and 2000 server products, which, unfortunately, are not free. If you are planning to receive a substantial amount of files, NLB probably isn't for you for reasons you'll see later in this chapter.

The major concerns you have with file access are typically the size of the files, which has the greatest impact on bandwidth, and multiple users accessing a file simultaneously, which results in loss of data integrity when changes are made.

Common Design Considerations for Special Media

Special media is a catch-all phrase that includes streaming audio and video and other similar data. These all share some pretty hefty requirements regarding their quality of service. As you learned in previous chapters, these include delay and jitter at the Network layer and the need for special **codecs** (short for *co*mpressor/*dec*ompressor) in the upper layers. Codecs translate audio or video from analog waves that humans hear into electronic signals.

Although a lot of solutions and workarounds to facilitate this type of traffic have been proposed, most are still in the development and testing phases, and only a few are widely implemented. The one solution that is gaining acceptance, however, is **IP multicast**. This technology allows a one-to-many connection at the Network layer, so that many clients can receive a transmission without requiring the sender to send a packet to each of them. This is historically a problem because of the time-sensitive nature of special media, such as voice and video. By the time the sender sends 5000 copies of a packet to 5000 clients, the next packet is already late, resulting in choppiness and clicking noises.

Enabling multicast on your site is a very advanced exercise, that involves a number of multicast routing protocols and special configurations for routers and switches, which are largely vendor-specific. So we'll only mention that IPv6 has much-improved support for these technologies and if it ever makes it to the Internet as we know it, it will probably stimulate a lot of enhanced and converged media applications.

While the networking infrastructure is complex, Microsoft has made the server configuration to support streaming media amazingly simple in Windows 2000 by using the

new Windows Media Service. This service runs on your server just like the WWW and FTP services, but it allows you to serve up content such as a live feed from a video camera or an MP3 file. The features in this service are very rich. They include the ability to provide broadcast or on-demand services, utilize unicast or broadcast traffic, and support almost any codec. Fortunately, they are relatively simple to configure. You can even choose from many different protocols. For instance, if there is a firewall in the path of your transmission that only allows HTTP traffic, you can configure the service to encapsulate its broadcast in HTTP.

After you have installed and configured it, the users of this system can access your content through Windows Media Player or Internet Explorer (IE). To use IE, you simply change the protocol type from http://<servername>/<resource> to mms://<servername>/<resource> in the URL field of IE. Using Windows Media Player is even simpler because you don't have to specify the protocol.

Of course, it is possible to create more complex situations in which you have many media servers involved, and you can separate their functions so that some are responsible for encoding the broadcast feed using a particular codec while others act as distribution points for content and can be spread around geographically. The permutations are endless. However, the caveat still applies: Before designing, installing, and configuring, you should understand the effect this traffic will have on your network and other applications.

Designing an Internet Site Infrastructure

Theory is great, but walking through the application of that theory is even better. To that end, this section illustrates basic design principles in action. We start with a minimal Internet site infrastructure design, which offers nothing more than basic service and connectivity, and then make changes to offer reliability, availability, security, and performance.

We start with the basic, essential elements:

- A server and OS
- IIS
- A network hub or switch (and assorted cabling)
- A router
- A CSU/DSU (for this example, we'll use a serial T-1 connection to the Internet)
- A connection to the ISP called the "local loop"
- An ISP

These components, as you probably know by now, fit together as shown in Figure 9-1.

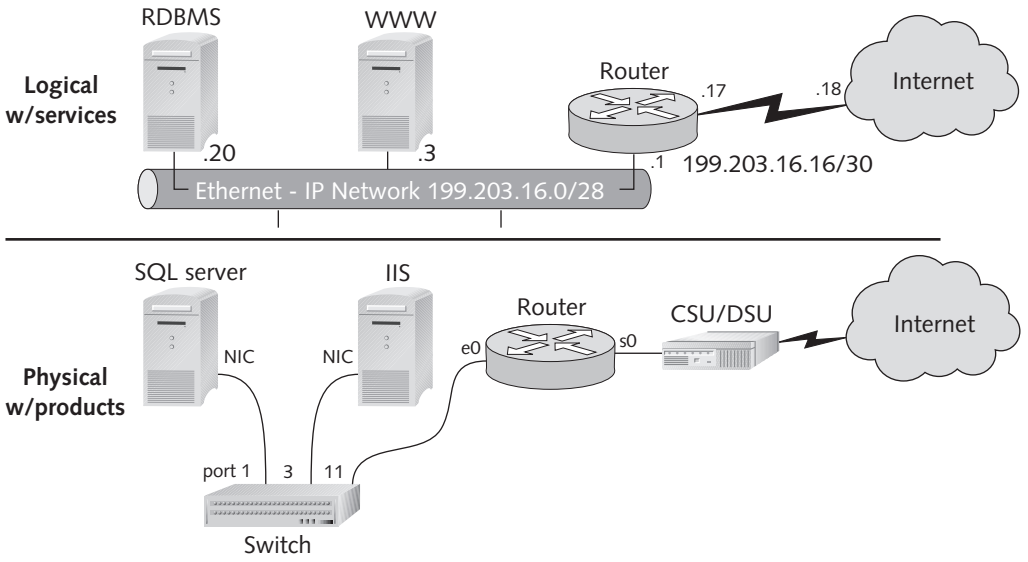

Figure 9-1 Minimum infrastructure required for connectivity

Single Point-of-Failure

The network in Figure 9-1 has some obvious issues. First of all, every single component in the network represents a **single point-of-failure**. That means if any wire gets a short, or if the switch, server, router, or ISP experience a failure, the entire service will be rendered inaccessible. Second, there is no security, other than whatever steps the administrator takes to make the Web and database servers more secure. This means that users on the Internet can access your database server directly, rather than using your Web application to access your data. This is typically a bad thing.

Our first improvement will be to separate the database server by creating two virtual LANs (VLANs) on the switch, as shown in Figure 9-2. To accomplish this, we must **dual-home** the Web server so that it has an interface on both VLANs. Also, we change the database server's IP address to one of the private ranges specified in RFC 1918—in this case, 172.16.200.20. Then, we remove the default gateway from the database server so that it cannot reach any networks that aren't connected.

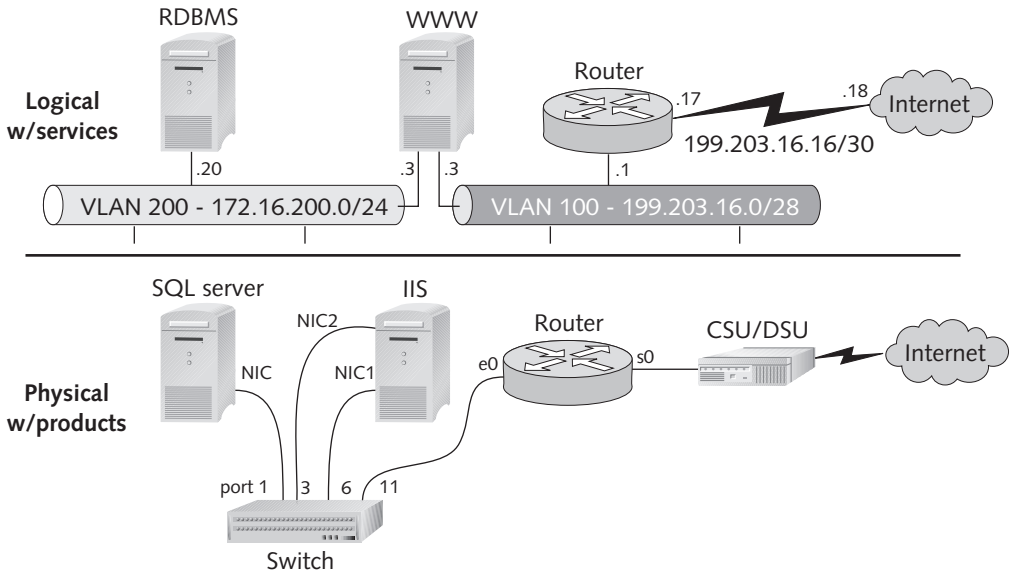

Figure 9-2 VLANs for database security

As you progress through these diagrams, you may notice that the IP addresses are not entirely random. Pay attention to the relationship between VLAN ID and the IP network. Although we didn't use color in this diagram (the publishing specs of this book didn't allow it), we recommend that you use colors to represent various objects, because as you will see, the diagrams can become quite cluttered with labels. Also, notice how similar server's IP addresses are grouped together. This requires a little planning in the design stage; otherwise, future expansion will cause your servers to break your scheme by having discontiguous addresses.

Redundancy

Now we're a little more secure than we were. However, we still have issues; so our next improvement in terms of cost-effectiveness will be to configure redundant connections to the Internet, as shown in Figure 9-3. We accomplish this by using two lines from our router to the ISP and making appropriate modifications to the routing protocol configuration. However, if you have more money and need more bandwidth, you can configure the same links to be load balancing, which also provides fault tolerance, instead of just redundant connections where you're paying for two links but only using one. (Presumably, the second link is much cheaper.)

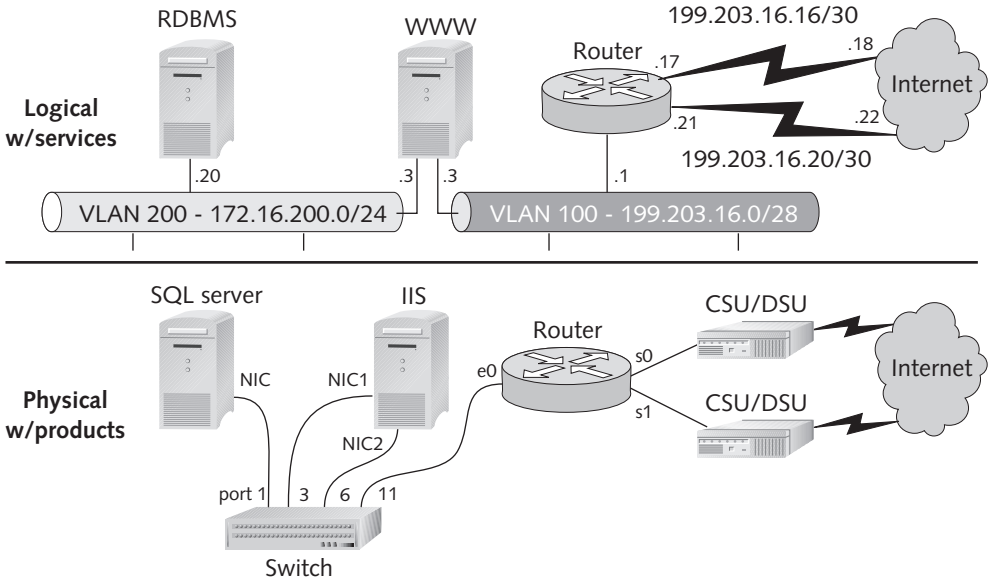

Figure 9-3 Redundant uplinks to ISP

Configuring multiple connections to our ISP will prevent a lot of outages, but we're still vulnerable to a hardware or software glitch in our router, switch, Web server, and database server. In Figure 9-4, we add redundancy to our router. This is also simple, but we want to point out some options here. Each router knows how to get packets from the Internet to the Web server, because it has only one address. However, how does the server know which router to send its responses to? The first option is to configure two default gateways. To do this, go to Network and Dial-up Connections in the Control Panel. Double-click your Local Area Connection associated with your Ethernet NIC. Next, double-click the TCP/IP protocol. You'll note there's only space for one default gateway here, but if you click the Advanced button, you'll be able to enter as many as you want.

In our example, we'll add a default gateway entry to 199.203.16.1 and another one for 199.203.16.2. If you're an old-timer, you also can use the command line interface with the following statements:

```
route ADD 0.0.0.0 MASK 0.0.0.0 199.203.16.1 METRIC 1
route ADD 0.0.0.0 MASK 0.0.0.0 199.203.16.2 METRIC 2
```

Note the different metrics used. IP devices will always prefer the lowest metric; so this configuration only offers redundancy and not load balancing. In theory, when Windows detects that the primary gateway is down by trying to resend a TCP packet half the times specified in TcpMaxDataRetransmissions, it will attempt to use the second gateway. The second option is to configure a routing protocol (either OSPF or RIP) on both the servers and the routers that will automatically detect and compensate for failures and that will potentially perform a load balance.

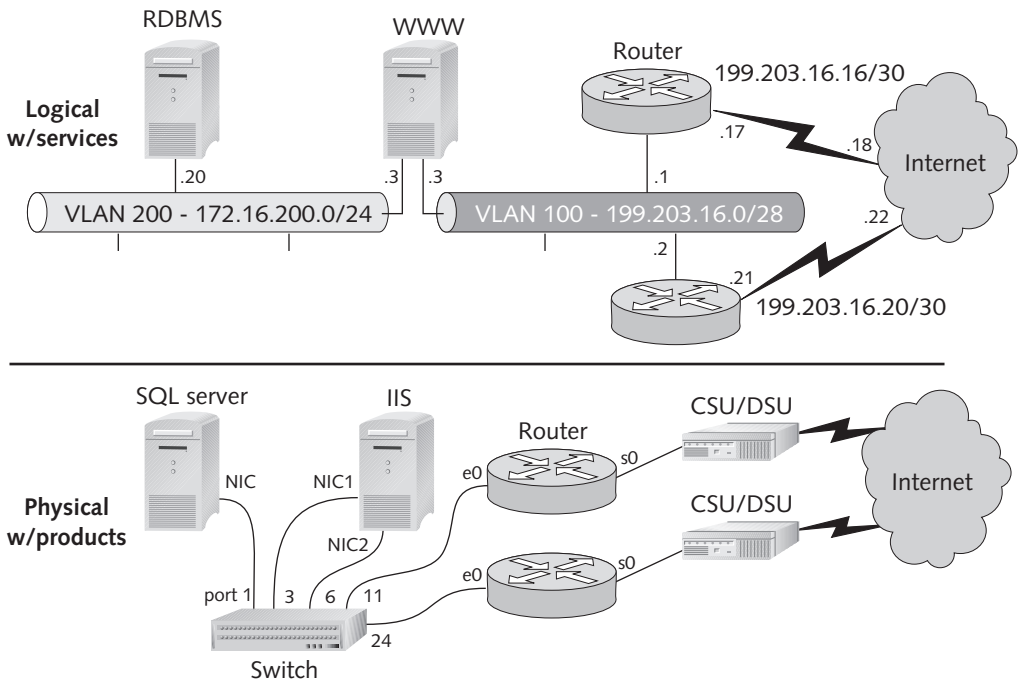

Figure 9-4 Redundant routers for the infrastructure

The third option is to use either **Virtual Router Redundancy Protocol (VRRP)** or **Hot-Standby Router Protocol (HSRP)** on the routers. They behave similarly, but VRRP is an open standard and HSRP is the more popular, Cisco proprietary protocol. Both offer a virtual IP address for hosts to send packets to. If the primary router becomes unavailable, the backup router will assume the virtual IP address so that no disruption in service occurs.

The Joy of Multiple ISPs

Now that our connection and routers are redundant, we'll also consider using multiple ISPs, as shown in Figure 9-5. This configuration can be important, because it is not uncommon for an ISP's entire network to fail. This may be caused by a software configuration error, routing loops, or a backhoe cutting a fiber cable, but the result is the same. If you decide to use multiple ISPs, have the ISPs review your network configuration to be sure you don't accidentally become a transit network between them.

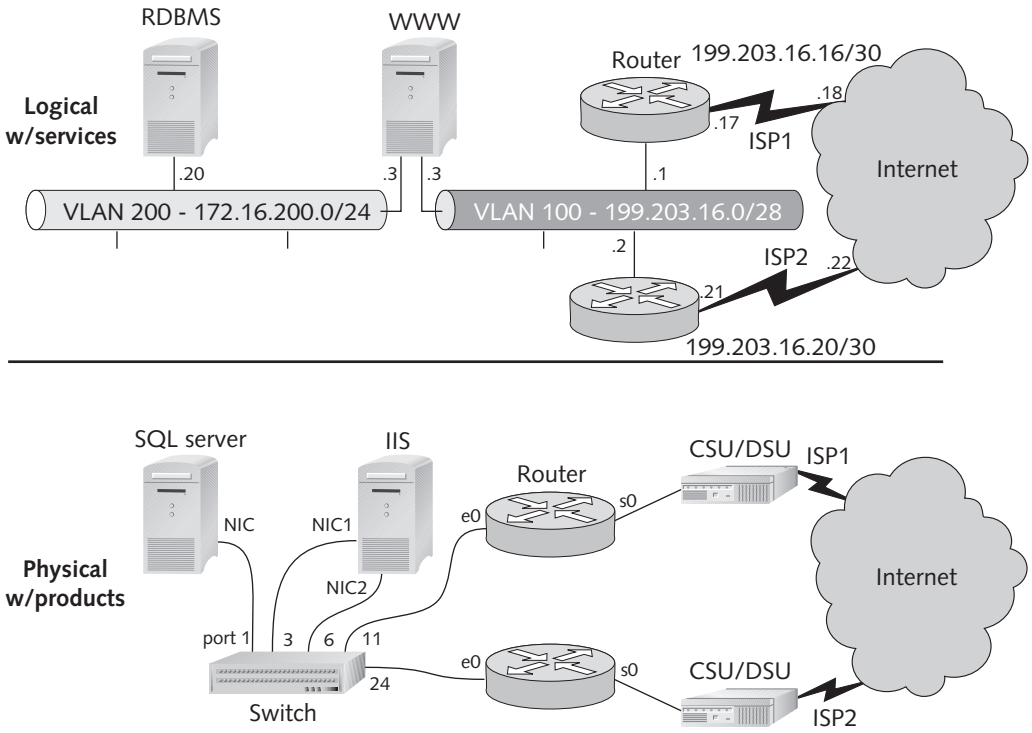

Figure 9-5 Redundant ISPs

Redundant Firewalls

Next, we're going to add a pair of redundant firewalls, as shown in Figure 9-6. For added security, we have opted not to use a third VLAN, but rather to physically separate the exposed network from the protected networks by using a hub. Also note that these firewalls are sharing an IP address much like the VRRP scheme we mentioned, in which all the traffic goes through one firewall. When it fails, the backup automatically assumes the IP address of the primary and prevents a service disruption.

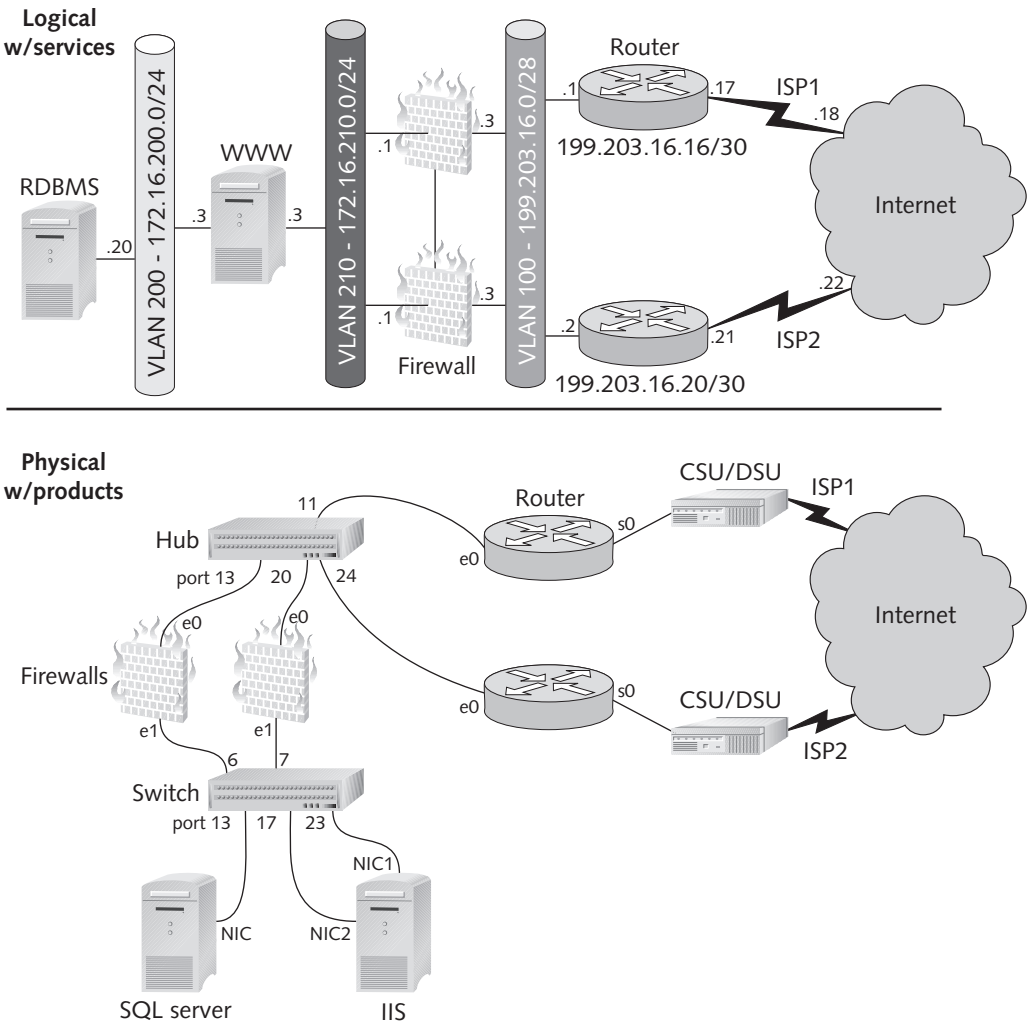

Figure 9-6 Redundant firewalls

If you're wondering why we used a hub here instead of a switch, that's a good question. There are several answers. First, there really is no significant disadvantage. Two 1.544 Mbps T-1 links, or even two 45 Mbps T-3 links, are not going to overload a 100 Mbps hub, and a few collisions aren't going to cause a substantial delay in the traffic. Second, the hub, as you know, sends all traffic to all ports. In most cases, like your standard office network, this is a security risk, but in this instance, it's actually a security benefit. A hub will allow us to insert a protocol analyzer for testing without disrupting service. It will also allow us to insert an **RMON (Remote MONitoring) probe** for monitoring utilization and other statistics. Last, it will allow us to insert an intrusion detection system to identify attacks as they happen and take steps to neutralize them.

With a hub, all three devices will see every single packet that crosses the wire, and they can be configured to notify you in the event of an emergency. Additionally, hubs are much cheaper than switches. If you do wish to use a switch, you may be able to use a feature called **port spanning** (Cisco terminology) or **port mirroring** (everyone else's terminology). Both allow your switch to make a copy of every frame that goes in or out of a set of ports that you designate and forward the copies out another port that you designate. Place your monitoring device off this last port so it can "see" everything. The advantage here is that your other Ethernet ports can remain in full duplex and not be bothered by traffic not destined for them. The disadvantage is that the span or mirror port has its Rx (or receive) circuit disabled. This means that your intrusion detection system will only be able to listen, not respond.

In this configuration, the Web server's IP address has changed to be in the private 172.16.210.0/24 network, and the default gateway on your Web servers has changed to the 172.16.210.1 address of the firewalls.

Most major firewall vendors offer quite a bit of flexibility in their implementation options. In this case, we have chosen to implement the firewalls where one firewall responds to all requests and the other sits and waits for the primary to fail. As it waits, it exchanges state and typically configuration information with the primary, so that when the primary fails, it can continue offering stateful features with no service disruption.

The backup firewall also maintains a heartbeat connection, in which it tests the primary firewall every few seconds (the number of seconds is configurable and the defaults vary by manufacturer), so that it can detect a failure and replace the primary. Instead of calling the second firewall a hot spare, most firewall manufacturers prefer the term **active-passive configuration**. This is opposed to an **active-active configuration**, where the firewalls load balance the traffic and exchange state information with each other so that both firewalls are capable of assuming the other's responsibilities in the event of a failure. In the opinion of these authors, the active-active configurations are impressive but overly complex to maintain and troubleshoot. The obvious reason for wanting an active-active configuration is twice the bandwidth, but it's much simpler and often cheaper to upgrade to Gigabit Ethernet NICs in your primary firewall than to use two active-active Fast Ethernet NICs.

Network Load Balancing (NLB)

In Figure 9-7, things begin to get complex (although we omitted some of the wires on the physical diagram for readability). Here we implement multiple Web and database servers. In a nutshell, Microsoft recommends Cluster Services for failover and for data changes that cannot easily be replicated. It also recommends **Network Load Balancing (NLB)**, which is a Microsoft product included in Windows 2000 Advanced Server that allows multiple servers to respond to requests in a way that is transparent to the user. You can use NLB for static or slow-changing data that can be replicated, and for automatic load balancing. So in this scenario, we'll use NLB on the Web servers and cluster the databases. For details on installing and configuring NLB, check out Microsoft Knowledge Base article Q240997, which you can access at *www.microsoft.com*.

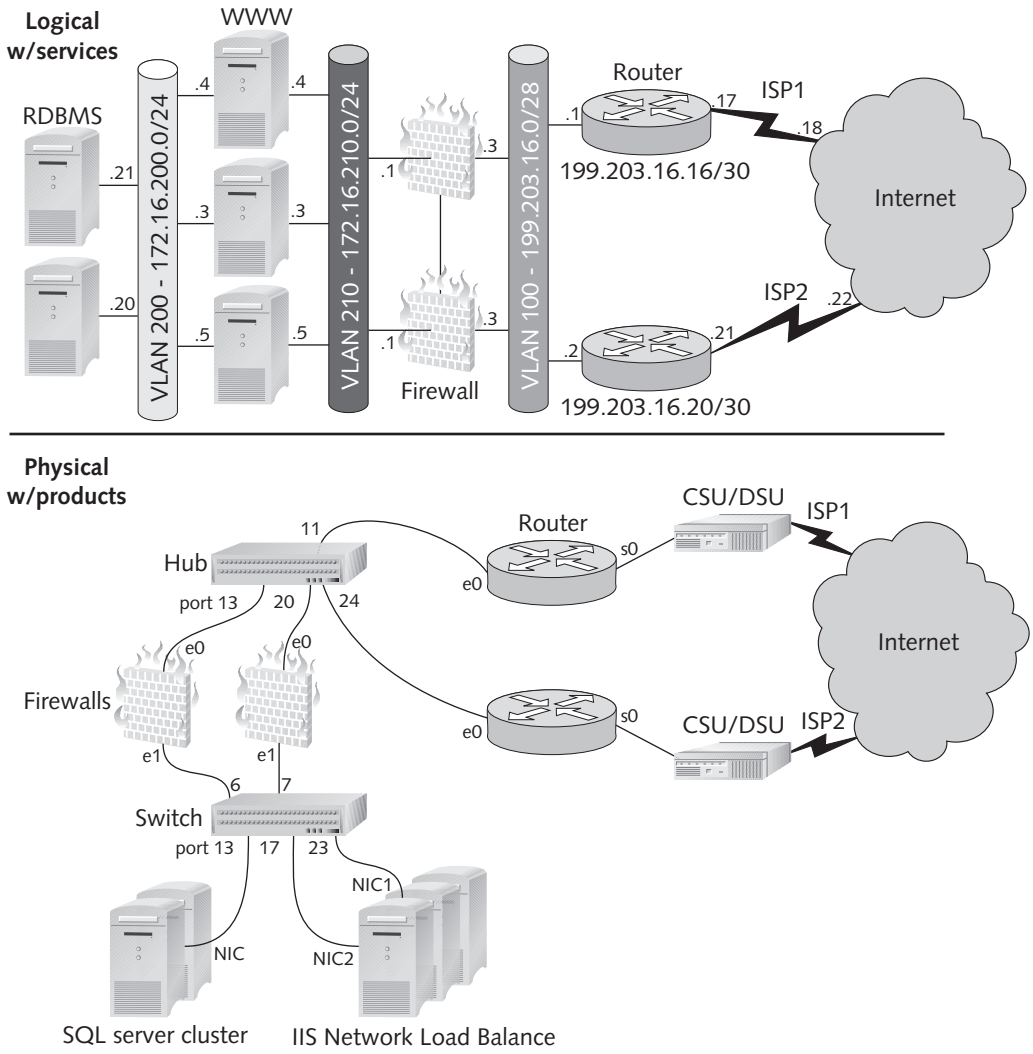

Figure 9-7 NLB for Web servers and Cluster Services for database servers

Although installing and configuring NLB isn't difficult, it is vital that you understand how it works when designing an infrastructure to support Web services using NLB. To illustrate the point, ask yourself these two questions:

- Would it make a difference if I use a switch or hub?
- Will traffic patterns change my design?

Let's look at the first question. Offhand, you might say there's no real difference in switches and hubs, except that switches allow you to use full-duplex connections and eliminate collisions. However, with NLB, there are some design issues to consider. These

issues exist because NLB operates by having all inbound traffic sent to all nodes in the subnet and then filtering unwanted packets on the servers. To do this, it has to configure a common MAC address and virtual IP address for each node in the cluster.

If the device in question is a hub, all traffic is sent to all ports anyway, and when it arrives at each station, each station will accept the MAC address and IP address as its own and pass the packet up the IP stack. But if the device is a switch, NLB has to trick the switch. Remember that switches aren't just fancy hubs; they're actually multi-port bridges, and in a normal network, when a layer 2 switch receives a frame from a port, it will take the source MAC address from that frame and enter it and the port number into its **forwarding database (FDB)**. When a frame is received from another port, it takes the destination MAC address and searches the FDB for a match. If it finds a match, it forwards the frame out the port number listed in the FDB entry.

All this is to say that if a switch received a frame with the MAC address of the cluster on a port, it would send all traffic destined for that MAC address out only that port. If that happens, NLB is broken, because only one actual node will receive frames. However, if it doesn't find a match, it **floods** the frame by sending it out every port in the switch, except the port it came from. So in this case, NLB isn't broken.

The problem now is that all MAC addresses are supposed to be unique. If the switch sees the same MAC address coming from two different ports (remember, NLB has assigned a common MAC address to every node in the cluster), it assumes it has an **Ethernet loop** where frames cycle endlessly around the network. So to resolve this situation, Microsoft again bends the rules by having each node in the cluster insert a different fake MAC address in the source address field of every Ethernet frame sent from the machine. This way, the switch never receives the MAC address of the cluster on any port and, therefore, will always flood frames destined to the cluster out every port in the switch so that all nodes in the cluster receive the frames and process them.

The danger here is that in an Internet environment, you might have a large switched infrastructure with many switches connected to each other. In this environment, your entire broadcast domain will receive every single packet (even though they're unicast packets and not broadcast packets) destined for your cluster. If you are doing some serious uploading to the cluster, you could easily choke your entire network by flooding these frames.

So what difference will a hub make? Well, if you directly connect all the nodes in the cluster to a hub, you can set the aptly named NLB registry entry MaskSourceMAC to 0 on all the cluster nodes, which prevents the bogus source MAC entries and thus allows switches upstream to enter the MAC in their FDBs and stop flooding the frames. The switches will forward the traffic to the hub, which will, in turn, propagate the frame out all ports.

Now if you return to Figure 9-7, you'll notice that we used a switch with our NLB cluster instead of a hub. This is because there aren't any other connected switches here, and the flooding will be restricted to the VLAN that contains the IIS servers, which is where we want the flooding to occur anyway.

As far as traffic patterns are concerned, this architecture operates under the assumption that request traffic sent to a server will be much smaller than the response sent from the server to the clients. In other words, all the traffic sent from the server to the clients is unicast and forwarded by the switches to a single destination, rather than flooded out all ports, while traffic from the clients to the server is flooded to all ports. Because requests are typically very small, it doesn't affect the network very much. However, if client-to-server traffic is substantial, then your network will be quickly overloaded.

How big is big? How substantial is substantial? Well, remember that NLB operates by having every single packet sent to all nodes in the cluster, but traffic from the cluster to the client is only sent by one node. If you take your switches, routers, uplinks, and ISPs out of the picture, your cluster can receive as much traffic as its slowest NIC. In addition, it can send traffic equal to all its NICs combined. For example, if you have four nodes in a cluster and are using Fast Ethernet NICs connected to a switch with a Gigabit Ethernet uplink, you could send 400 Mbps (100 Mbps × 4 NICs) but only receive 100 Mbps.

Solutions That Compete with NLB

Now that you have a conceptual grasp of Microsoft's Network Load Balancing, we feel this is an appropriate place to discuss some competing solutions. Microsoft typically refers to these solutions as dispatcher-based because they intercept requests to your cluster and simply swap the original MAC address in the frame with the MAC address of the server to which they have decided to send the frame.

Dispatching behavior has pros and cons. On the one hand, it can make intelligent decisions based on combinations of factors, from CPU utilization on the server, to current number of requests, to which server has a particular page. Essentially, you get a more useful load balancing not just based on the number of requests, but also based on the server conditions. They also can test to make sure the applications are responding instead of just making sure the server's NIC is still responding. By dispatching, you only send each request to a single server so that you don't flood your network, and in terms of bandwidth, you're only constrained by the interfaces on your load balancer. However, as Microsoft points out, the actual decision making involved in dispatching will add a little delay to the flight time of your packets. In addition, dispatcher-based solutions are often quite expensive. Because of that, for reliability, availability, and performance, you'll need at least two, and maybe several.

Because most of these solutions are hardware-based and all the vendors have slightly different design preferences, we chose not to include a sample diagram using hardware-based load balancing. Instead, our next upgrade will be the establishment of a connection from your office network to the Web site. This connection will support server administration, content upgrades, and connectivity from your site's database and Web servers to your enterprise information systems.

Figure 9-8 shows the additions we've made to the logical and physical diagrams to connect this site to your corporate network. This addition is fairly simple and adds another NIC to every server in the farm and another switch, a terminal server, and a firewall.

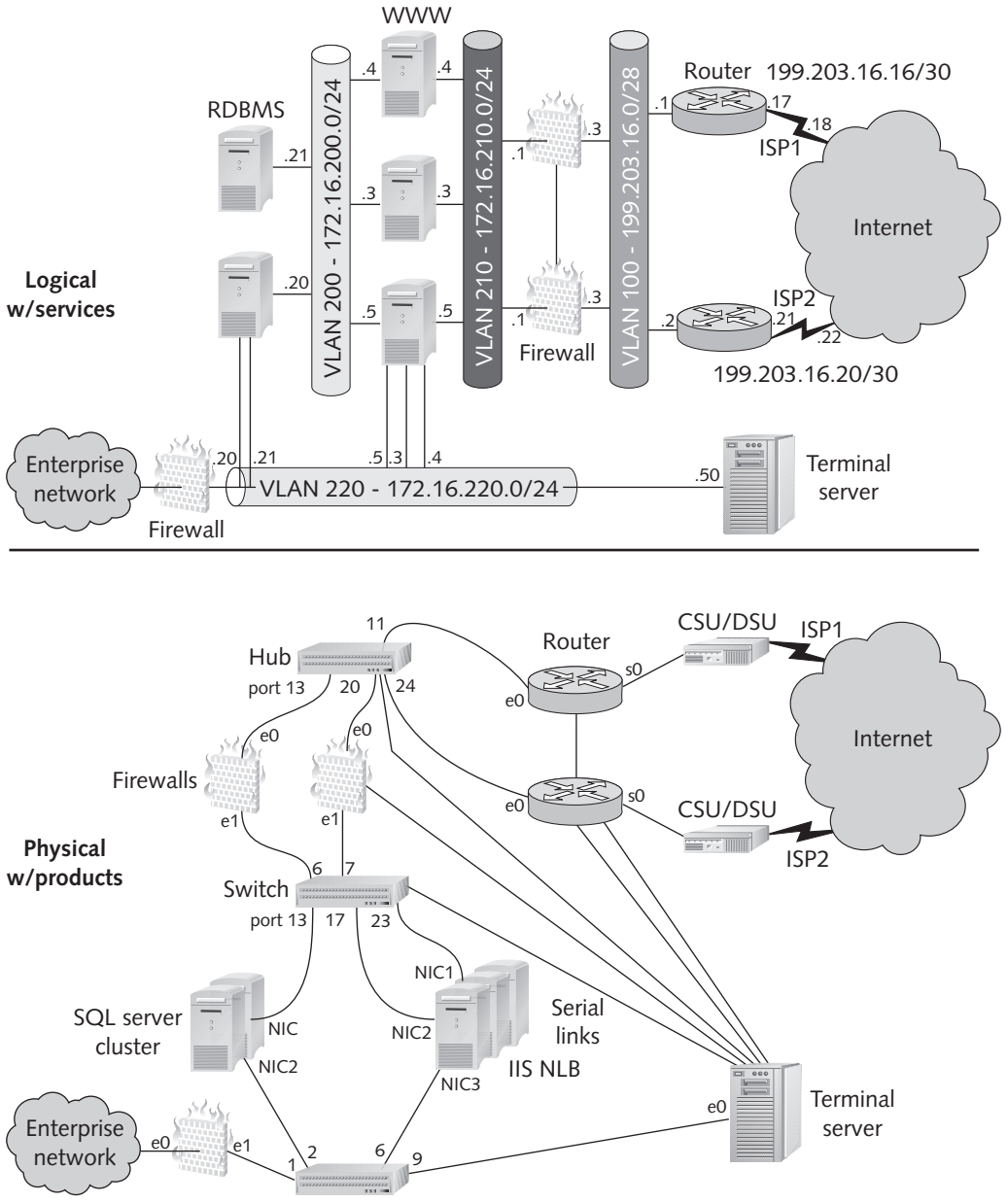

Figure 9-8 Back-end and management network

There are a couple reasons why we chose to add separate NICs instead of using the existing ones. The first is a technique called **out-of-band management**. It means having a dedicated network for management. Often this is done to keep from having your management traffic contend with your user data traffic. In this case, we don't want the traffic between the database and Web server to be affected by our management activity. However, since we are

planning on running data from the mainframe on our corporate network across this link, it is technically an **in-band management** link. It also offers a little extra security.

When installing these NICs, pay special attention to your TCP/IP properties. You need a way to tell the server how to reach your corporate network, and you don't want to use a default gateway; otherwise, the administration traffic would return through the Internet, or the HTTP responses from the servers to Internet clients could return through your corporate network. The two preferred ways to accomplish this are by configuring a static route to your network management console on each server or by running NAT on the back-end firewall. If you run NAT on the back-end firewall and assign a static entry for your network management station and any servers or hosts that the server farm needs to talk to, the entry will be on the local IP network and no route will be required. This also affords a little more protection and flexibility.

The terminal server is used to provide a console connection into all the routers, switches, and firewalls. The console connections are typically through a serial link (like a modem cable) and because they're dedicated and don't run IP, they're much more secure than just using a LAN interface and Telnet. Because Windows servers don't have console connections (they have a monitor, keyboard, and mouse instead), you'll have to find another solution to remotely manage those.

Switch and Hub Redundancy

At this point, we want to take a few steps back and start adding some redundancy into the switch and hub. Adding redundancy at layer 2 can be extremely confusing, because of the limitations of common server hardware. The primary goal here is that in the event of a failure in a hub or switch, connectivity won't be lost. If possible, the secondary goal here is to prevent loss of bandwidth or performance. In the real world, designers often add a second, active connection for redundancy, but as time passes, the utilization grows to the point that the second line is needed for capacity. Unfortunately, the engineers rarely realize that this is no longer redundant until it's too late.

We start the change by adding a second hub, which we connect to the first hub through a crossover cable or an uplink port, as shown in Figure 9-9. This means that devices on one hub can communicate with the other. Ideally, we would like to connect both routers to both hubs, so if one hub fails, both routers are still active. However, most routers won't allow this. If they have two Ethernet interfaces, they typically won't allow you to put both interfaces in the same IP network. So we're forced to connect one router to each hub. This means that if one hub fails, that router will be unreachable, which in turn means that we lose half of our very expensive bandwidth to the ISP.

To handle this situation, we have also connected the two routers' second Ethernet interfaces with a crossover cable, but these interfaces, each labeled e1 on Figure 9-9, will be in a different IP network and we'll route between them. Remember that when you configure two equal-cost default routes on each router, with one pointing toward the Internet and the other pointing toward the other router, you can have a hub failure without taking out half your bandwidth. If your router fails, though, you're still out of luck.

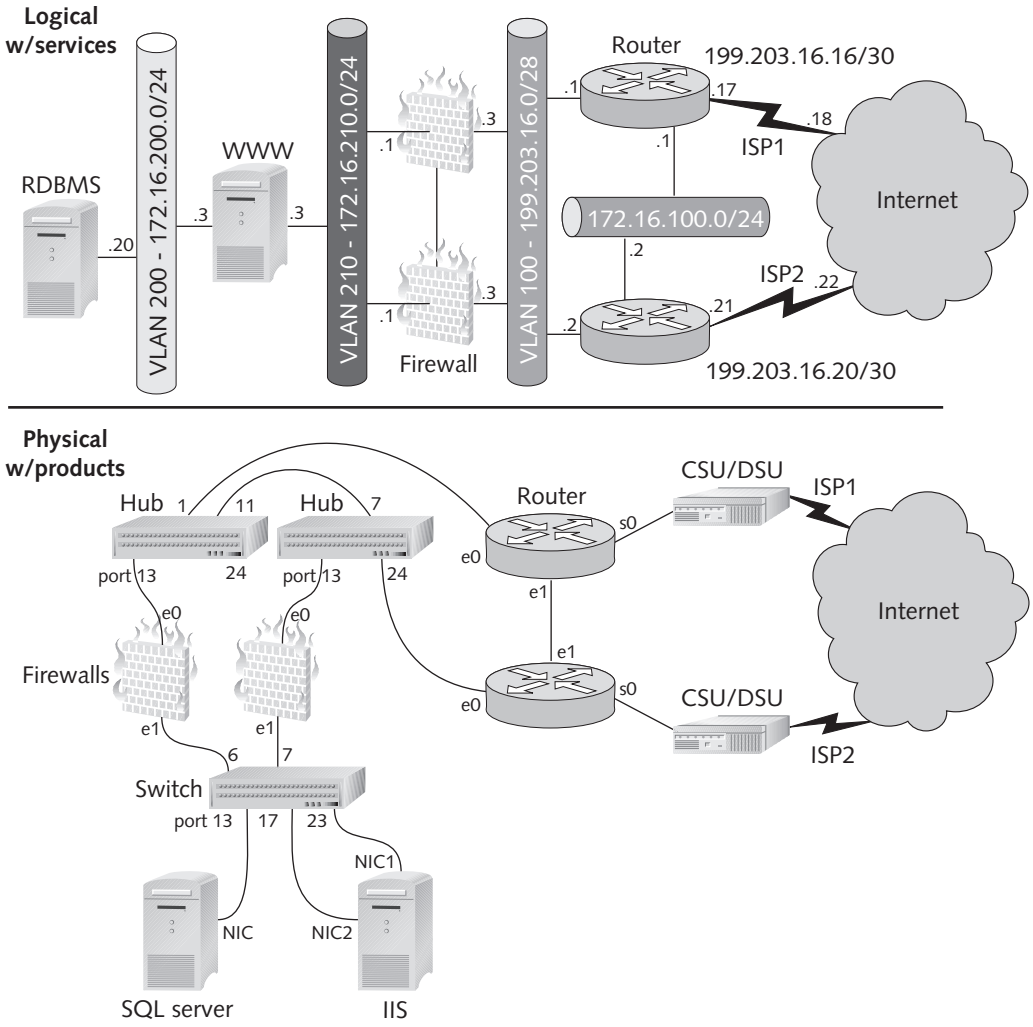

Figure 9-9 Hubs for uplink protection and network visibilty

Firewalls

Now let's look at the firewalls. Recall that our firewalls are in what's called an active-passive configuration, where only one is responding to requests and forwarding packets, while the other is hibernating. Although you want to plug them into separate hubs for redundancy, it doesn't really matter which hub the active firewall is plugged into as long as you are load balancing your links to the ISP. If you have a primary and backup link to the ISP, you'd obviously want to plug the active firewall into the same hub as the primary router. As long as the throughput of the firewall is more than the aggregate throughput of the uplinks to the ISP, bandwidth isn't really a concern.

Redundant Switches

In Figure 9-10, we will implement redundant switches. This is complicated for a number of reasons. First, there are multiple VLANs. In addition, you need VLAN 200 and VLAN 210 on each switch, and you need connectivity between corresponding VLANs on each switch. You could, of course, front-panel connect them by stringing a crossover cable connecting a port in VLAN 200 on one switch to a port in VLAN 200 on the other switch, and then repeat this for each VLAN. However, this has several major disadvantages. First, each VLAN will use up two ports. Second, it's ugly. This may sound superficial, but when you start troubleshooting, having a bunch of wires that go from one switch to another can be very confusing because you can't tell what VLAN they're in. It also can be confusing if you accidentally grab a straight-through cable instead of a crossover.

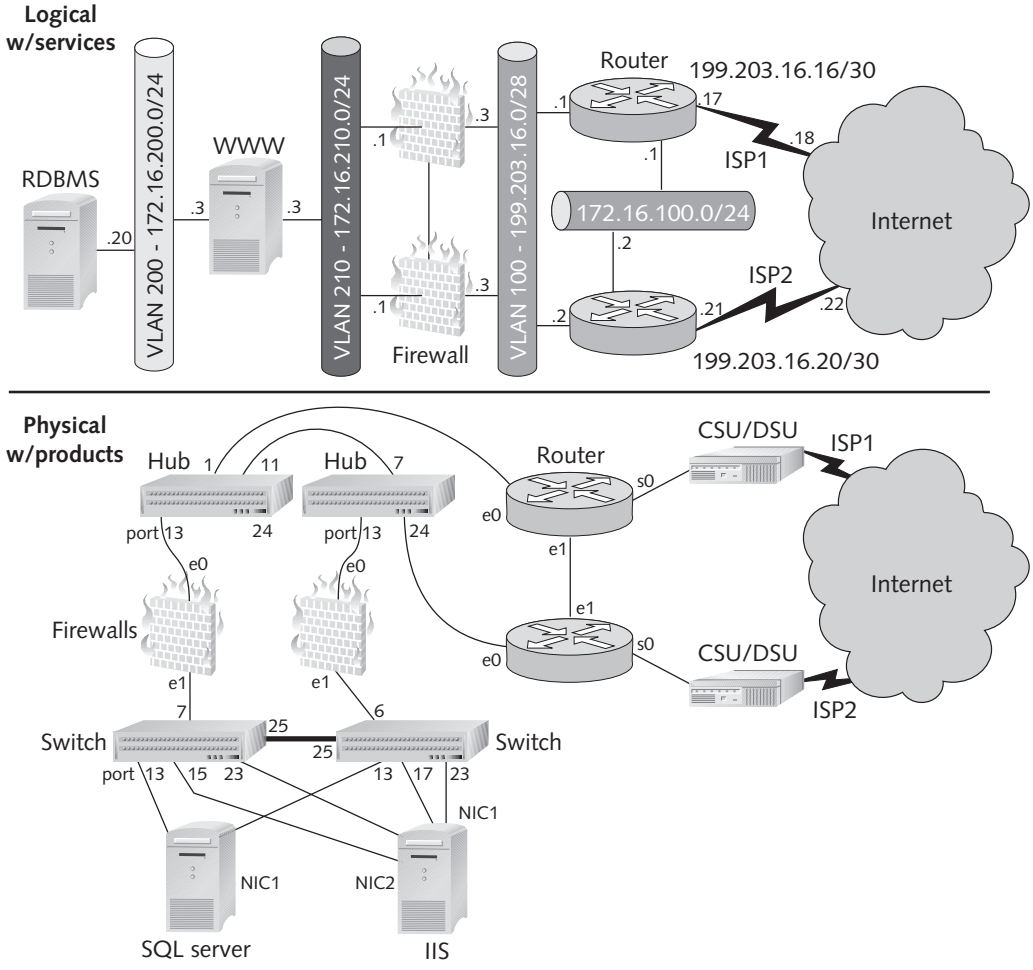

Figure 9-10 Redundant switches and VLAN trunking

The practice of plugging patch cables from one switch into another can easily lead to the dreaded mistake of connecting two cables in the same VLAN—otherwise known as an Ethernet loop, or a bandwidth-reduction scheme, as depicted in Figure 9-11. When a host sends a broadcast, it will be forwarded out all ports, including the two crossover links to the other switch. When the other switch receives the frame on port 15, it will forward it out all ports except the one it came in on, which includes port 23! Meanwhile, it also receives the same broadcast frame (but doesn't realize it's the same) from port 23, which it forwards out all ports except 23, which includes, of course, port 15! When these two frames get back to the first switch on ports 3 and 11, the two frames are sent right back out ports 11 and 3, respectively. You can see where the term "Ethernet loop" comes from.

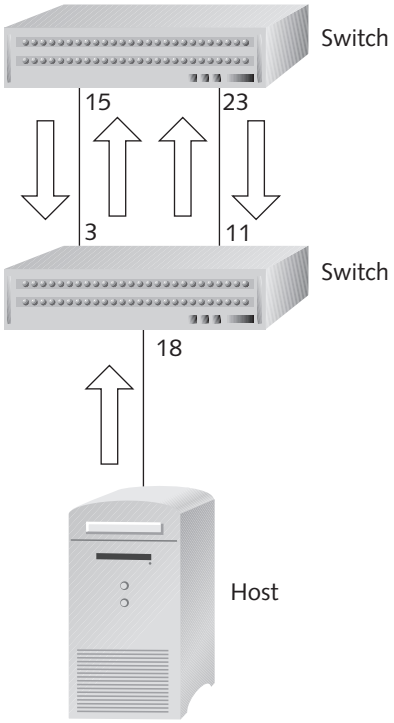

Figure 9-11 Ethernet loop

In the Ethernet loop, the two frames will continue circulating around your switches until one of those two links is unplugged. In this situation, don't forget that every host on this network gets a copy of both broadcasts, every cycle. Eventually, enough broadcasts will accumulate to use up all available bandwidth. This is affectionately known as a "reduction scheme." The only redeeming feature of an Ethernet loop is watching the lights on your switches flash really fast, if you're into that sort of thing.

Now that you understand part of the problem, the two simple solutions are to *not* front-panel connect your switches or to run the **Spanning Tree Protocol (STP)**. The STP is important to layer 2 redundancy because it was designed by IBM decades ago for exactly that purpose—to allow multiple bridges to be connected without creating loops. It accomplishes this by stopping all user traffic, sending Bridge Protocol Data Unit (BPDU) frames out each port, and running a little algorithm. If it finds a loop, it sets one of the ports into a blocking mode, which simply means it doesn't transmit and only receives BPDUs. It then places all the other ports into a forwarding mode.

In the event the active uplink fails, the STP will shut down all the ports again and repeat the process, which results in the other link being made active, because the loop no longer exists. In a large network, this process can take more than a minute and occurs every time a change is made to a switch, which can be highly annoying to say the least. However, in a typical Internet site, the process should complete in a few seconds and occur very rarely. So other than the nominal BPDU traffic, STP is a good thing and an integral part of most redundancy schemes.

The other component that can be used here is called **VLAN Trunking**. This technology allows multiple VLANs to be connected through a single physical cable, which can really save a lot of ports if you have several VLANs. You'll especially want to use this if you're connecting your switches together with more expensive and lower-density Gigabit Ethernet ports.

One popular protocol that you would typically consider is IEEE's **802.1Q**, which puts a 4-byte tag in the Ethernet header of each frame. This tag includes a VLAN ID field so that the receiving switch knows which VLAN to send the frame to. The other protocol is Cisco's **Inter-Switch Link (ISL)**, which actually encapsulates the entire Ethernet frame by placing a header and checksum around it. ISL also includes a VLAN ID field, which performs the same function.

It is worth mentioning here that even though our switches are redundant now, that really only means we have protected them from link failures and hardware failures, and by using spanning tree, we have prevented Ethernet loops. All of this is layer 1 and 2 protection. As you can imagine, that is not an exhaustive list of things that can go wrong on a network. For instance, what if layer 3 fails? An event like a broadcast storm from another source (not just an Ethernet loop) could deny service to all network interfaces in that broadcast domain. If you had 100 clustered servers with dual NICs on this network, during a broadcast storm you'd have 200 NICs not responding to your requests! For this reason, extremely redundant networks often employ a mix of failover features at all layers of the OSI model.

Generally speaking, faults are corrected much faster at lower levels of the OSI model. For example, redundant hardware often has sub-second failover time. STP can take a few seconds to a few minutes, and routing protocols can take several seconds or several minutes to reconverge. Likewise, a fault-tolerant layer 2 protocol such as LLC2 will detect and resend a packet much faster than TCP at layer 4, because the time-out period is necessarily longer the higher up the OSI model you go.

Some Final Thoughts on Switch Failures

Now that we've established connectivity between switches, we still have a problem: Our servers have only one NIC (per VLAN); therefore, a switch failure will take out half our servers. The obvious solution of adding another NIC is generally not recommended, as it tends to confuse some applications and protocols. A better solution is a dual-port NIC that almost all server hardware vendors offer. These NICs allow you to run a cable to each switch.

We can configure one port as the primary port and the other as a backup. The advantage of using a dual-port NIC is that the driver software has some very convenient features: When the primary fails, it will bring the secondary link up and spoof the MAC address and IP address of the primary. This scheme can achieve sub-second failover.

As a final note on switching, as long as our firewalls are in active-passive mode, we want our primary server NICs to all be connected to the switch used by the active firewall. In active-active mode, it's somewhat more complex, but generally preferable to balance the load by making one switch the primary for half your servers and backup for the other half, and have the other switch act as a backup for the first half and as a primary for the other half.

It is extremely important to pay attention to which layer you're providing redundancy on. The dual-port NIC operates at a Physical layer; so the only change in the network that is required at a higher layer is the backup switch that is putting a new entry into its forwarding database. STP operates at layer 2, and layer 3 and up should be completely oblivious to any faults that occur. On the frontend of the site, the routers implement fault tolerance at layers 2 and 3 by using their routing tables and VRRP/HSRP. In this case, we could have designed many different solutions, but we chose these to illustrate the way fault-tolerant solutions operate at all the different layers.

The Nuances of Providing Services to an Intranet

Enterprise intranet sites are very similar to their Internet counterparts in that they use the same protocols and services, but there are a few differences worth noting. The first is that they generally reside on the same LAN as the bulk of the users; so applications are often implemented with no regard for bandwidth.

The second difference we'll mention is that intranet sites typically support a wider range of applications. Historically, the intranets of many companies began on the desktops of IT employees and then evolved into departmental Web sites for sharing information. At some point, some small applications are developed to enhance productivity or automate some workflow tasks, and gradually the intranet site becomes mission critical. Larger applications such as ERP programs are added, and TN3270 gateways to the mainframe allow Web browsers to replace terminal software on the clients. In addition, many applications with often drastically different requirements are implemented in a single site.

The third difference is that security is woefully inadequate and often nonexistent on most intranet sites because of the assumption that access is limited to employees and, therefore, security is unnecessary. Contributing to this is a lack of resources. While high-profile Internet sites get proportionally high-profile budgets, intranet site security is the first "nonessential" item in a budget to be cut. Nevertheless, security either is or is not a design requirement for your site. If you determine that it is a requirement, but the budget gets cut, be prepared to pursue less expensive methods of mitigating your security risks, rather than simply declaring that security is no longer a requirement. This also should be well documented.

Providing an Infrastructure for Services to an Intranet

As we turn our attention to intranet infrastructure, we again look at some of the considerations that shape our design and how these are different from the prevailing Internet considerations. Then, to give you a real-world application of your newfound knowledge, we explore the nuances of providing your intranet services to the Internet. In the process, we take a look at two important tools: screened subnets and reverse proxy servers.

Design Considerations by Type

In this section, we look briefly at some design considerations by content type. As you read this section, keep in mind that intranet designs can't be done in a vacuum. All the considerations we mention here are related to your internal and external user and server environments. If you don't take the existing environment into account, you can expect to encounter glitches, where performance becomes an issue or things simply may not work because of incompatible technology.

Design Requirements for Enterprise Applications

The term **enterprise application** refers to any of the mission-critical programs that run your business. Examples would be any of the Enterprise Resource Planning (ERP) packages or software that takes input from a customer interface and then sends instructions to computerized equipment in your factory or distribution centers.

Most of the design requirements that apply to enterprise applications are protective in nature. Their basic goal is to prevent other, less important applications from trampling on your critical ones. For instance, if you were running IIS with an FTP server on the same server that had your enterprise application, you wouldn't want user downloads to restrict this traffic. One example of a feature designed to prevent this is **bandwidth throttling**, as implemented in IIS 5.0. This feature prevents other applications from consuming all the bandwidth by allowing you to specify the maximum transmission rate in Kbps.

In Hands-on Project 9-3, you'll see how to configure rate limiting.

Design Requirements for File Access

Unlike Internet file-sharing sites, where people might download MP3s or programs and then leave, users of intranet sites will often access the site continuously for eight hours per day. Between the more evenly distributed load and the known audience, you can more accurately estimate the hardware and network size requirements for file access to your intranet site.

File transfers are typically constrained by the speed of the storage media, the server's bus architecture, and the speed of the network. Unfortunately, most intranet content is much too volatile to be suitable for replication and thus NLB.

Another thing to consider here is that while Internet sites are typically done by a small group of dedicated people, intranet sites are often done in a very distributed fashion, with a person or two from each department in the company contributing or managing content. Consider the merits of integrating Active Directory if your environment resembles this.

In addition, in an intranet environment, you need to be aware that the same file could be retrieved by a standard Windows NetBIOS file server request, through a Web server, or through an FTP server if you set the share, wwwroot, and ftproot directories to be the same. While admittedly this sounds more like a feature than a concern, and you're not likely to see this on the exam, most experienced network administrators will tell you that offering too many choices to users can be a bad thing. Six months down the road, it will be time to migrate some users' PCs and you'll find that some of them have shortcuts on their desktops, others have macros using UNC paths, others have mapped drives, and still others have the URLs in their Favorites menu or cache, and if you miss any of that, there will be much complaining.

Design Considerations for E-Mail

Intranet e-mail is very different from its Internet counterpart. The challenges here are rarely volume or bandwidth, but compatibility and how many programs can interact. Sometimes it seems like every Windows-based program now has some feature that takes advantage of the Outlook client or Exchange server. These programs typically use e-mail as a transport to offer a collaborative enhancement to another application. For instance, Microsoft's Project 2000 includes a Web server that allows project information to be stored and managed through a Web browser on an intranet site. This solution allows resources to e-mail their hours worked and task status to the project server, so that the project can be updated. Your update also can include information from your Outlook calendar so that Project can create a more accurate schedule. As this kind of application becomes increasingly common and increasingly intertwined, e-mail and all its associated services in the intranet becomes a mission-critical part of the infrastructure.

Designing an Intranet Site Infrastructure That Offers Services to the Internet

The infrastructure for intranet sites differs from that of Internet sites mostly in scale but little in concept, except in the lack of uplinks to an ISP and the common practice of putting all the servers, databases, and so forth on the same IP network. So instead of spending a great deal of time in this chapter on the connection between your internal users and the intranet, we want to explore a special circumstance in which you want to make the information on your intranet available to selected users on the Internet.

In this case, we'll assume that your internal network is already in existence and your intranet site, including IIS and SQL Server, are on the same physical and logical network as your internal users. Now obviously, we don't want to allow access from the Internet to these servers directly because the potential for security issues is simply too great, even with a stateful firewall in place. So what can we do? The example solution we'll propose here takes advantage of two concepts: **reverse proxy** and a **screened subnet**.

Reverse Proxy and Screened Subnet

Regular proxy service forwards requests on behalf of its users. In other words, it intercepts a request (for example, an HTTP request for a Web page), and then scrubs the request and contacts the Web server, pretending to be a client. Once it has retrieved the object, it responds to the client, pretending to be a Web server. In this manner, the client is anonymous, and therefore more secure. Also, it can improve performance because objects are cached, which reduces the network traffic on the Internet because the proxy server only needs to retrieve items once and all the clients using the proxy can access those items.

A reverse proxy performs a similar function, except that it sits next to the Web server, and DNS points to the proxy server instead of the actual Web server, so all requests go to the proxy. Then queries are passed to the Web server and a response is sent from the proxy to the real clients across the Internet. The benefit here is that the Web server is protected by anonymity. Also, if the Web server isn't overly powerful, the reverse proxy also can cache pages and improve response times.

The proposed solution will protect our Web server, but alas, we don't want to put the reverse proxy server on the Internet network for the same reason we don't want Internet users accessing the Web server on the internal network. Thus, our solution is to create a third network, called a screened subnet. See Figure 9-12.

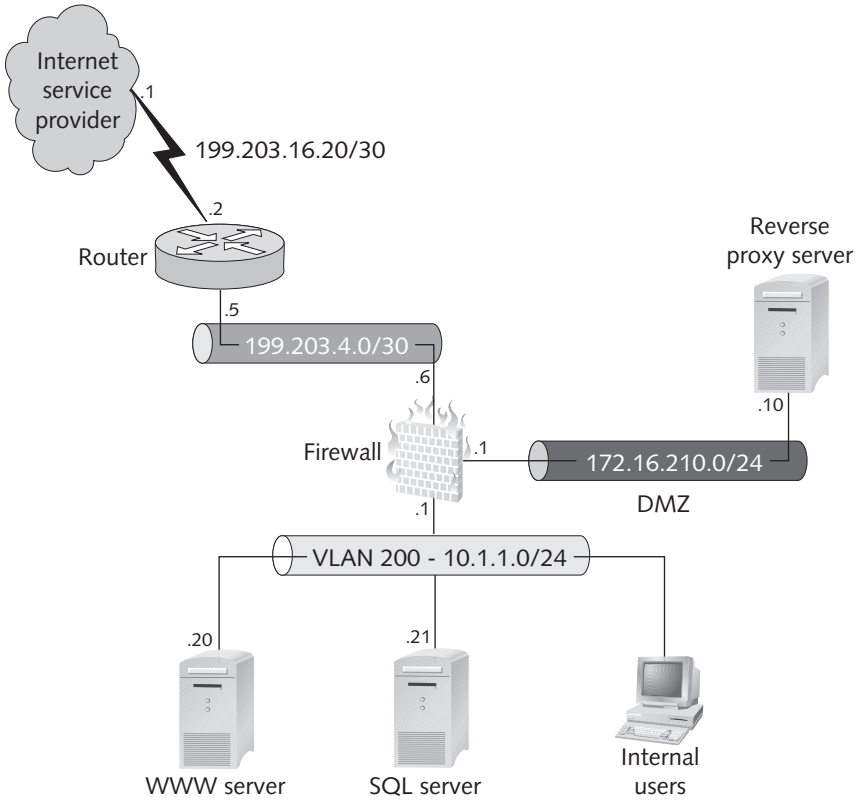

Figure 9-12 Screened subnet and reverse proxy server

Don't let the term "subnet" fool you. It could be a classful network or a subnet of a classful network.

Note that a screened subnet is not an unprotected network, where anything can pass, and it is not a protected network, where nothing can pass. Instead, access is permitted only through a firewall, which only permits certain traffic into the network. A screen subnet is sometimes called a DMZ because it is a middle ground or "no man's land." This implies that it is not completely unprotected, but is less protected than your internal network. In practice, the firewalls generally block all ports except the ones required for the service—for example, ports 80 and 25, if you wanted to host e-mail and SMTP. In addition, firewalls provide some protection against DoS and other security attacks.

So to access the intranet, a user on the Internet will request the IP address from their DNS server and receive the IP address for the reverse proxy server. Not knowing the difference, the user's computer will attempt to send an HTTP request to the reverse proxy server. The firewall will allow a TCP connection to port 80 to be established and

the proxy will then receive the request and attempt to forward it to the intranet Web server. Because the firewall has been configured to allow traffic only from the IP address of the reverse proxy server into the internal network, the firewall will allow a TCP connection between the Web server and reverse proxy server to be established and the Web server will receive and respond to the request. The reverse proxy will then forward the response to the client and tear down both TCP connections. The next time the reverse-proxy server receives a request for the same object, it will simply respond from its cache, rather than request the item from the Web server again.

This reverse proxy functionality is available in Microsoft's Proxy Server 2.0. Also, as we mentioned earlier, Microsoft's ISA product is too new to be covered on the exam, but it offers much additional functionality for intranet site design that was not previously available. Be sure to research these products thoroughly before completing your design.

Special Considerations When Designing Internet and Intranet Sites

In the world of networking, there are many rules that must be followed to make the network "work." Occasionally, however, we decide that the advantages of breaking a rule outweigh the disadvantages. In this section, we discuss several technologies that break the rules and what effects these may have on your network.

Challenges with Load Balancing

We've already covered the nuts and bolts of Windows Network Load Balancing and explained the alternate dispatcher solutions; so we know that there are many advantages to load balancing, including scalability, availability, and reliability. The difficulty with load balancing, particularly the dispatcher-based solutions, is that the packets' headers are altered as they travel through the network. This affects protocols that store the original IP or MAC address in the data portion of the packet. It also affects load balancing, which can potentially redirect your traffic from one server to another during conversation. IPSec is a protocol that might be broken because it stores the IP address in the data portion of the packets. If you feel you need both a dispatcher-based solution and IPSec, make sure that the solution you choose only alters the MAC address in the frames and not the IP address.

Web developers' applications typically rely on something called state. **State** is a way for the application to know to whom it is talking. Many Internet sites consist of a series of pages where you enter information and make choices, like the ubiquitous shopping carts. To remember which user filled out what, Web servers typically use either session variables or cookies. Although Microsoft's NLB handles these automatically, most dispatcher solutions have to jump through some hoops to make sure they deliver your packets to the same server your shopping cart is on. If the load balancer supports this feature, it's typically done by reading deep into the data field of the packet and tracking

the same information that the Web server does. Then the load balancer can always send your packets to the same server. This is known as a **sticky connection**. We cannot overemphasize the importance of testing these products before deploying them.

For more information on maintaining state, visit *http://msdn.microsoft.com/workshop/server/feature/webfarm3.asp.*

One last consideration when determining a load balancing solution is the use of proxy servers and DNS. Since most load balancing is done based on source IP address, and all traffic that passes through a proxy server is altered to swap the source address of the client with the IP address of the proxy server, your load won't be evenly distributed. In a typical B2C site, you can usually trust statistics to even things out, but if half your users are from one large company, no matter how many servers you have, all their traffic will be sent to a single server.

DNS has a similar problem, in that many people try to load balance their traffic by putting IP addresses in DNS for each server. This scheme is called "round-robin DNS", because if the DNS server has two IP addresses associated with a FQDN, when the DNS server receives a request, it will respond with the first IP address on its list. The second request gets the second address on its list, and the third request will be sent the first address on the list again.

With round-robin DNS, traffic is balanced by requests and an individual will use the same connection until the DNS cache on the computer expires and has to resolve the name again. However, large ISPs and many corporations run caching DNS servers. Let's assume one of your customers has 500 clients that use your servers. This means that the first person from that customer who sends a DNS request will receive an IP address, but that IP address (and only that IP address) will be stored in their corporate DNS cache, so everyone else who sends a request will get the same response from their DNS server instead of your DNS server. This also can skew the load on your servers and defeat your load balancing scheme.

A site is said to be data-driven when its pages are largely rendered based on information that is contained in a database. This is opposed to static pages that always render the same way, or mostly the same way. An example of a data-driven site would be *www.ebay.com*, where all the pages except the first page look different every time you click the Refresh button because the contents of the database are constantly changing. While this certainly doesn't break any rules, it can cause you to look for some creative solutions for scaling your back-end or database connections. These solutions, such as caching and load balancing, will break the rules and generally require some application-level work to compensate.

The job of a firewall in an intranet site is to stop unauthorized traffic from reaching your servers. A packet filtering firewall does this by dropping packets based on source or destination IP address, or various other fields in the layer 2, 3, and 4 headers. As the bad

guys get sneakier, the firewalls have to get more sophisticated. Thus, a stateful firewall knows all the networking rules and attempts to make sure everyone else follows them, while it breaks the rules itself by dropping packets selectively. However, because it knows how the protocols operate, and not just the contents of a packet, it can drop a packet based on the contents of other packets. For instance, to establish a TCP connection, which is required by the HTTP protocol, your PC has to send a SYN packet to the server. The server responds with a SYN-ACK packet, and your PC responds with a FIN packet. This is known as the three-way handshake.

Of course, what happens if someone decides to just send you a SYN packet? And another? And another? Each of these will eat up your server's resources and eventually deny service to others. A packet-filtering firewall would let each of these through, while a stateful firewall would let a couple through, and then realize something is amiss and deny the rest. Another advantage of stateful firewalls can be seen when someone sends you a FIN packet without first sending you a SYN and SYN-ACK packet. A stateful firewall would realize that no SYN or SYN-ACK had been received and deny this FIN packet, which could thwart a number of attacks. As you can see, stateful firewalls can be very useful.

However useful they are, firewalls do break some rules, and occasionally there are consequences. One consequence is when firewalls are load balanced in high-traffic sites. If the SYN packet comes in one firewall and the SYN-ACK packet goes out another, and then the FIN packet comes in through the first firewall, it will be dropped because the first firewall never saw your server's response. Of course, this means your TCP connection will never be established and the client will return a time-out message. If your firewalls are configured in an active-active mode, they may exchange this state information; so this isn't a problem. Nonetheless, make sure you test your configurations thoroughly.

Network Address Translation

Very few Internet sites employ NAT on hosts that can be reached from the Internet, simply because it's not necessary and it breaks a lot of other protocols. Another reason is that if both ends of a connection are using NAT, things can get really tricky. In addition, even though it's a software feature, it represents a point of failure. A glitch in the software or an error in the configuration can cause a lot of problems for a technology that doesn't offer a lot of advantages to Internet sites. Intranet sites, on the other hand, are often already using private RFC 1918 addresses and the clients and servers are often on the same network; so no translation is necessary, or even possible.

You might want to employ NAT in an Internet site if your environment is highly volatile. If you plan to swap a lot of servers in and out of service, it could be a pain to change all their IP addresses and DNS entries. Instead, NAT would provide a layer of abstraction, where you could simply change the static entry in NAT and avoid changing host IP and DNS entries altogether.

Chapter Summary

- In this chapter, we looked at the infrastructure required to support common Internet services. We discussed common design considerations for categories of sites such as B2B and B2C and how they're different. We also discussed requirements and considerations for the types of traffic, such as e-mail, file access, and special media.
- We then walked through the design phase of an Internet and intranet site and explained the reasons behind many design decisions and what popular alternatives are available. This led us through a detailed discussion of load balancing, clustering, firewalls, and proxy technology and solutions. We also discussed the merits of implementing redundancy and failover at different layers of the OSI model.
- We then looked at some of the considerations that shape our design of providing services to an intranet and how these are different from the prevailing Internet considerations. Specifically, we concentrated on design requirements for enterprise applications, file access, and e-mail. Then we explored the nuances of providing your intranet services to the Internet. In the process, we looked at two important tools: screened subnets and reverse proxy servers.
- Last, we explored the situations in which breaking a rule outweighs the disadvantages. This discussion concentrated on the technologies that break the rules and the effects these "rule-breaking" situations have on your network.

Key Terms

802.1Q — IEEE's open protocol specification for VLAN tagging.

active-active configuration — A pair of devices that are configured in such a way that both are in service simultaneously and if either fails, the other will assume its role.

active-passive configuration — A pair of devices that are configured in such a way that only one is active and if it fails, the other will assume its role.

B2B — A corporate Internet presence that allows companies to transact business or share information with each other.

B2C — An e-commerce or informational site that is used by consumers.

bandwidth throttling — A condition or configuration that limits the rate of transmission, usually described in Kbps or Mbps.

batch-oriented processing systems — A computing methodology where work is queued and several units of work are processed at a time. This is easy to program and configure, but it imposes artificial delays.

blackhole service — Compiles lists of servers known to originate spam.

Blowfish — A fast, free encryption scheme. For more information, visit *www.counterpane.com/blowfish.html*.

brute force — A method of solving a problem by trying all possible combinations as quickly as possible, rather than reverse engineering.

codecs (short for *co*mpressor/*dec*ompressor) — Translate audio or video from analog waves that humans hear into electronic signals.

cold spare — A device that is not currently in service but can be manually placed in service to replace an identical device in the event of a failure.

computationally secure — A method of encryption that is more expensive to break than the encrypted data is worth, or that takes so long to break, the data would be worthless by the time it is broken.

cookie — A small file that is used to track user information and state, which is stored on client computers by Web browsers.

Data Encryption Scheme (DES) — A common 56-bit encryption scheme. For more information, visit *www.itl.nist.gov/fipspubs/fip46-2.htm*.

Denial of Service (DoS) — A security attack that prevents the use of a service.

digital certificates — An electronic file that confirms an identify. Specified in X.509, it contains a name, serial number, expiration date, and the public key to be used for encryption.

digital signatures — A method of identifying a message that provides nonrepudiation.

dual-home — A network that has multiple connections to the Internet or a server that has multiple connections (NICs) to a network.

Electronic Data Interchange (EDI) — A standard format for exchanging business data. The standard is ANSI X12.

enterprise application — Any of the mission-critical programs that run your business.

Escon — A channel-based, Data Link layer technology used by IBM mainframes.

Ethernet loop — A condition where frames in an Ethernet are endlessly forwarded in circles.

event-driven — A computing methodology where events can be defined, and those events can act as triggers to initiate responses.

Extended Binary-Coded Decimal Interchange Code (EBCDIC) — A binary code for alphabetic and numeric characters that IBM developed for its OS/390 operating system.

floods — Occur when a switch sends a frame out all ports except the port that the frame arrived on. This usually happens when the switch does not have an entry for the destination MAC address in its forwarding database.

forwarding database (FDB) — A database in layer 2 devices that matches a port with a MAC address. Used to determine where to send frames.

hot spare — A device that is configured to assume the responsibility of an identical device with no human intervention.

Hot-Standby Router Protocol (HSRP) — A protocol for IP gateway failover specified in RFC 2281 but primarily used by Cisco.

in-band management — Describes a condition when the traffic in question is configured to use the same connection as other data traffic.

Inter-Switch Link (ISL) — A Cisco proprietary specification for VLAN tagging.

IP multicast — A technology that allows a one-to-many connection at the Network layer, so that many clients can receive a transmission without requiring the sender to send a packet to each of them.

Network Load Balancing (NLB) — A Microsoft product included in Windows 2000 Advanced Server that allows multiple servers to respond to requests in a way that is transparent to the user.

nonrepudiation — Occurs when someone sends you a message (a transaction, for instance) and you can prove that they sent it.

out-of-band management — A condition when the traffic in question is configured to use a dedicated connection so as not to interfere with other data traffic.

port mirroring — A common industry term used to describe a configuration that sends copies of frames from one or more ports to a designated port. Used for protocol analyzers, RMON devices, and so on.

port spanning — A Cisco term used to describe a configuration that sends copies of frames from one or more ports to a designated port. Used for protocol analyzers, RMON devices, and so on.

proof-of-concept — A scaled-down version of an entire project.

reverse proxy — A service that allows a proxy server to respond to all user requests on behalf of the actual server.

RMON (Remote Monitoring) probe — A subset of the SNMP protocol that provides history, statistics, alarms, and so on for network traffic.

screened subnet — A network that is exposed to another organization but partially protected by a firewall.

security token — A device that provides a special key required to log on to a system.

single point-of-failure — A physical or logical object in a system whose failure will cause the entire system to fail.

spam — Unsolicited or unwanted e-mail messages.

Spanning Tree Protocol (STP) — An IBM protocol adopted by IEEE that configures a group of switches to prevent loops.

state — Maintaining information about a user during a visit to a Web site, or maintaining information about a stream of data in the network.

steganography — The practice of concealing a message.

sticky connection — When traffic directors send all packets from a single source to a single destination.

strong authentication — An authentication scheme that combines multiple schemes, typically requiring a user name, password, and either a token or certificate.

Systems Network Architecture (SNA) — An architecture created by IBM for communications between hosts.

Triple DES — An encryption scheme similar to DES that uses 128 bits.

unconditionally secure — An encryption scheme that cannot be broken because the information required to unlock the encryption is not transmitted with the message.

Virtual Router Redundancy Protocol (VRRP) — An open protocol for IP gateway failover used by many vendors.

VLAN Trunking — A network configuration that allows traffic from multiple VLANs to be transmitted and received on the same interface. Used to conserve expensive uplink ports.

Review Questions

1. If you wanted to secure the integrity of your data as it is transmitted across the Internet, which of the following would you use?
   a. encryption
   b. strong authentication
   c. checksums
   d. digital certificates
2. Which applications are appropriate for Network Load Balancing?
   a. Exchange Server
   b. SNA Server
   c. IIS — WWW
   d. IIS — FTP uploads
   e. IIS — FTP downloads
3. What will happen if your Windows 2000/IIS server does not have a default gateway configured?
   a. It can receive and send packets to the Internet.
   b. It can receive packets, but it can only send to its local network.
   c. It cannot receive packets, but it can send packets.
   d. It can neither receive nor send packets.
4. STP operates at which layer of the OSI model?
   a. layer 1
   b. layer 2
   c. layer 3
   d. layer 4
5. How does Windows 2000 Server behave if two default gateways are configured with the same metric?
   a. It will load balance by alternating packets between the two gateways.
   b. It will load balance by alternating destinations between the two gateways.
   c. It will result in a routing loop and no packets will be sent.
   d. It will send all packets to the first gateway.

6. How does Windows 2000 Server behave if two default gateways are configured with different metrics?
   a. It will load balance traffic between the two gateways proportional to their metrics.
   b. It will send all packets to the gateway with the lowest metric.
   c. It will send all packets to the gateway with the highest metric.
   d. None of the above. Metrics only affect inbound traffic.
7. How many MAC addresses does a switch store in its FDB for an NLB cluster?
   a. one for each node in the cluster and one for the cluster's virtual IP
   b. one for the cluster's virtual IP
   c. one for each node in the cluster
   d. none, because the MAC addresses are never sent
8. If seven servers are used in an NLB cluster and all are configured with 100 Mbps Fast Ethernet NICs, what is the total bandwidth of the cluster?
   a. 700 Mbps in and 700 Mbps out
   b. 100 Mbps in and 350 Mbps out
   c. 700 Mbps in and 100 Mbps out
   d. 100 Mbps in and 700 Mbps out
9. Which layer of the OSI model offers the fastest recovery time for failover?
   a. layer 1
   b. layer 2
   c. layer 3
   d. layer 4
10. How many servers does Microsoft Cluster Services support?
    a. two
    b. three
    c. four
    d. five
11. Which of the following can make an encryption scheme computationally secure?
    a. third-order mathematics
    b. time required to crack
    c. money required to crack
    d. domain-key form

12. If server A receives inbound HTTP packets from client B through one stateful firewall and sends outbound HTTP packets to client B through another stateful firewall, what symptoms might you expect?
    a. Both firewalls will drop packets.
    b. The second firewall will redirect them through the first firewall.
    c. unbalanced access bits set in the IP headers
    d. a routing loop between the firewalls
13. Single points of failure are _____.
    a. better than dual points of failure
    b. always bad
    c. bad only if your design requirements say they're bad
    d. system bottlenecks
14. Bandwidth throttling refers to _____.
    a. limiting the number of TCP/IP connections
    b. limiting the number of packets per second
    c. limiting the number of Kbps
    d. slowing the transmission speed of packets on the wire
15. What consideration is crucial when implementing SMTP services?
    a. whether NAT is used
    b. which TCP port the SMTP client uses
    c. which users are allowed access to the server
    d. whether you have an SMTP-compatible Ethernet NIC
16. What port in your firewall will you want to open for inbound traffic when hosting a Web server?
    a. TCP port 19
    b. TCP port 35
    c. UDP port 159
    d. TCP port 80
17. If you have four ISPs connected to two routers, connected to one firewall, and connected to a server, how many default routes should you configure on the server?
    a. one to the firewall
    b. two to the routers
    c. four to the routers
    d. four to the ISPs

18. What version of Windows 2000 is required to support NLB?
    a. Windows 2000 Professional or Server
    b. Windows 2000 Server or Advanced Server
    c. Windows 2000 Advanced Server or Datacenter
    d. Windows 2000 Datacenter only
19. On an out-of-band management network, what percent of total traffic should user traffic not exceed?
    a. 0 percent
    b. 20 percent
    c. 50 percent
    d. 80 percent
20. Another name for a screened subnet is _____.
    a. tunnel
    b. DMZ
    c. VPN
    d. NACK
21. How far into a header must a layer 3 switch read before making a decision?
    a. the Ethernet header
    b. the IP header
    c. the TCP header
    d. the UDP header

HANDS-ON PROJECTS

Project 9-1 Configuring Default Gateways on Windows 2000

You will need a computer running Windows 2000 that is configured with TCP/IP and administrative rights. You also will need a host that will respond to ICMP pings on another network and an IP address that is not currently in use on your network. To configure multiple gateways on Windows 2000:

1. If your server is not powered up, power it up now. If it is powered up, skip to Step 7.
2. Press **Control/Alt/Delete** to display the Log On to Windows dialog box.
3. In the User Name text box, type **administrator**.
4. In the Password text box, type **password**. (If this does not work, ask your instructor for the password.)

5. In the Log on to text box, use the selection arrow to select **INTERSALES**. (This will depend on the classroom configuration.)
6. Press **Return**.
7. When the desktop appears, click the **Start** button on the taskbar.
8. Click **Run**.
9. In the Open text box of the Run dialog text box, type **cmd**.
10. Click the **OK** button.
11. At the command prompt, type **ROUTE PRINT** and verify that there is only one route with destination to 0.0.0.0 and netmask 0.0.0.0.
12. Type **PING *x.x.x.x***, where *x.x.x.x* is the address of a host on another network that will respond to pings. Verify that you receive successful replies; the output from the PING command should say "REPLY FROM" four times.
13. Point to the **Start** button on the taskbar.
14. Point to **Settings**, and then click **Network and Dial-up Connections**.
15. Right-click **Local Area Connection**, and then click **Properties**.
16. Highlight **Internet Protocol (TCP/IP)**, and then click **Properties**.
17. Click **Advanced**.
18. Click the **Add** button under Default gateways.
19. Type an address into the Gateway field that is not currently in use by any other computer on your subnet. Ask your instructor for an address, if necessary.
20. Type **1** into the Metric field.
21. Click **Add**.
22. Click **OK** three times.
23. Open the Command Prompt window from your taskbar.
24. Type **ROUTE PRINT**, and then verify the change to the routing table. You should see a default route to the IP address you configured.
25. Type **PING *x.x.x.x***, where *x.x.x.x* is the address of a host on another network that will respond to pings. Verify that you receive successful replies.
26. Open the Network and Dial-up Connections window from the taskbar.
27. Right-click **Local Area Connection** and then click **Properties**.
28. Highlight **Internet Protocol (TCP/IP)**, and then click **Properties**.
29. Click **Advanced**.
30. Highlight your original default gateway, and then click **Edit**.
31. Change the metric to **2**.
32. Click **OK** four times.
33. Open the **Command Prompt** window from your taskbar.

34. Type **ROUTE PRINT**, and verify the change to the metric.
35. Type **PING *x.x.x.x***, where *x.x.x.x* is the address of a host on another network that will respond to pings. Verify that you do *not* receive successful replies. You should *not* see a "REPLY FROM" in the output of the PING.
36. Open the Network and Dial-up Connections window from your taskbar.
37. Right-click **Local Area Connection**, and then click **Properties**.
38. Highlight **Internet Protocol (TCP/IP)**, and click **Properties**.
39. Click **Advanced**.
40. Highlight the default gateway you added, and then click **Remove**.
41. Click **OK** three times.
42. Close all open windows on your desktop.
43. Observe a moment of silence in honor of Windows 2000 and be thankful you no longer have to reboot your computer every time you make a change to your network settings.

Project 9-2 Restricting Access to IIS 5.0 Web Services

You will need a computer running Windows 2000 that is configured with TCP/IP and IIS 5.0 and administrative rights. The default configuration should be installed with the WWW service. To restrict access to the Web server:

1. If your server is not powered up, power it up now. If it is powered up, you may skip to Step 7.
2. Press **Control/Alt/Delete** to display the Log On to Windows dialog box.
3. In the User Name text box, type **administrator**.
4. In the Password text box, type **password**. (If this does not work, ask your instructor for the password.)
5. In the Log on to text box, use the selection arrow to select **INTERSALES**. (This will depend on the classroom configuration.)
6. Press **Return**.
7. When the desktop appears, click the **Start** button on the taskbar, point to **Programs**, point to **Administrative Tools**, and then click **Internet Services Manager**. You should see your computer listed in the left pane with an asterisk next to it.
8. Right-click your computer, and then click **Properties**.
9. Next to Master Properties, click **Edit**.
10. Click the **Directory Security** tab.
11. Click **Edit** under IP address and domain name restrictions.
12. Click **Denied Access**, click **Add**, and then click **Group of computers**.
13. Enter your network address and mask.
14. Click **OK** four times.

15. Try to access the Web service from a computer on your network.
16. Try to access the Web service from a computer on a different network.
17. Close all open windows.

Project 9-3 Configuring Bandwidth Throttling

You will need a computer running Windows 2000 that is configured with TCP/IP and IIS 5.0 and administrative rights. The default configuration should be installed with the WWW service. To control the transmission rate:

1. If your server is not powered up, power it up now. If it is powered up, you may skip to Step 7.
2. Press **Control/Alt/Delete** to display the Log On to Windows dialog box.
3. In the User Name text box, type **administrator**.
4. In the Password text box, type **password**. (If this does not work, ask your instructor for the password.)
5. In the Log on to text box, use the selection arrow to select **INTERSALES**. (This will depend on the classroom configuration.)
6. Press **Return**.
7. When the desktop appears, click the **Start** button on the taskbar.
8. Point to **Programs**, point to **Administrative Tools**, and then click **Internet Services Manager**. You should see your computer listed in the left pane with an asterisk next to it.
9. Right-click your computer, and then click **Properties**.
10. Check the **Enable Bandwidth Throttling** check box.
11. In the text box labeled Maximum Network Use, type **56**.
12. Click **OK**.
13. Close all open windows.

Project 9-4 Installing Media Services

You will need a computer running Windows 2000 that is configured with TCP/IP and you'll also need administrative rights. To install Windows Media Services:

1. If your server is not powered up, power it up now. If it is powered up, you may skip to Step 7.
2. Press **Control/Alt/Delete** to display the Log On to Windows dialog box.
3. In the User Name text box, type **administrator**.
4. In the Password text box, type **password**. (If this does not work, ask your instructor for the password.)
5. In the Log on to text box, use the selection arrow to select **INTERSALES**. (This will depend on the classroom configuration.)

6. Press **Return**.
7. When the desktop appears, click the **Start** button on the taskbar.
8. Point to **Settings**, click **Control Panel**, double-click **Add/Remove Programs**, and then click **Add/Remove Windows Components**.
9. Scroll to the bottom and check the **Windows Media Services** check box.
10. Click **Next**.
11. At this point, you may be prompted to make decisions about other components residing on your server. Click **No** or **Cancel** to all such dialog box prompts.
12. Click **Finish**.
13. Close all open windows.

Project 9-5 Configuring Media Services to Broadcast Files

You will need a computer running Windows 2000 that is configured with TCP/IP and IIS must be running with its default configuration using c:\inetpub\wwwroot. You'll also need administrative rights and you must have completed Hands-on Project 9-4. To begin configuring Windows Media Services:

1. If your server is not powered up, power it up now. If it is powered up, you may skip to Step 7.
2. Press **Control/Alt/Delete** to display the Log On to Windows dialog box.
3. In the User Name text box, type **administrator**.
4. In the Password text box, type **password**. (If this does not work, ask your instructor for the password.)
5. In the Log on to text box, use the selection arrow to select **INTERSALES**. (This will depend on the classroom configuration.)
6. Press **Return**.
7. When the desktop appears, click the **Start** button on the taskbar.
8. Point to **Programs**, point to **Administrative Tools**, click **Windows Media**, click **Multicast Stations**, click the **Stations** list arrow, and then click **New**.
9. Click **Next** five times.
10. Type **sample.asf** in the Source URL text box after the name of your server.
11. Click **Next**.
12. Type **C:\ASFRoot\sample.asf** in the Path text box.
13. Click **Next** four times.
14. Click **Finish**, and then click **Save**.
15. Click **Test .asx**. You should see your Windows Media Player attach to your computer.
16. Close the Windows Media Player when finished.

Project 9-6 Limiting Bandwidth on Broadcast Files

You will need a computer running Windows 2000 that is configured with TCP/IP and IIS must be running with its default configuration using c:\inetpub\wwwroot. You'll also need administrative rights and you must have completed Hands-on Project 9-5. To limit the bandwidth used to transmit files:

1. If your server is not powered up, power it up now. If it is powered up, you may skip to Step 7.
2. Press **Control/Alt/Delete** to display the Log On to Windows dialog box..
3. In the User Name text box, type **administrator**.
4. In the Password text box, type **password**. (If this does not work, ask your instructor for the password.)
5. In the Log on to text box, use the selection arrow to select **INTERSALES**. (This will depend on the classroom configuration.)
6. Press **Return**.
7. When the desktop appears, click the **Start** button on the taskbar.
8. Point to **Programs**, point to **Administrative Tools**, click **Windows Media**, and then click **Unicast Publishing Points**.
9. Click the **Broadcast** list arrow, and then click **Properties**.
10. Change Maximum Bandwidth from no limit to **Limit to**.
11. Press **OK**.
12. Close all open windows.

CASE PROJECTS

Case 9-1 When a Ping is Successful But a Copy Fails

While attempting to copy files between two nodes of an NLB cluster in unicast mode, you notice a bizarre problem. Your Ping appears to be successful, but the file copy fails. You check both servers and everything appears normal. You try to copy the files to a file server that isn't in the cluster and it works fine. You then copy the files from the file server to the other cluster server and that also is successful. You attach a protocol analyzer between the two servers and ping again. The server says the ping is successful, but the analyzer does not capture the packets. What is going on here? (*Hint:* Diagram the network described here. Make up IP addresses and MAC addresses for the servers and cluster and label them on your diagram.)

Case 9-2 Working Through an Upgrade

Your company has 200 users and an intranet site that contains several applications that are vital to your business, but the site is hosted on a single server with no redundancy. After explaining the situation to management, they have decided to put $65,000 in next quarter's budget for upgrading the intranet, and they want you to propose a design immediately.

Your applications are Web-based, but they interact with large, shared Access database files that are accessed through a share on a server. Your design needs to make this server redundant as well, and both existing servers are very old and should be replaced completely. All users currently reside on the 192.168.1.0/24 network and do not have Internet access.

After shopping around, you find the following equipment available:

- Server hardware/software: $14,000
- 10/100 Mbps Ethernet Switches: $2000
- Load balancer hardware/software: $25,000
- Router with two 100 Mbps Fast Ethernet interfaces: $6000
- Firewall with two 100 Mbps Fast Ethernet interfaces: $8000

You have plenty of spare cables, NICs, hubs, and other miscellaneous items and do not need to include those in your budget. Design the most redundant network possible without exceeding the budget given. Draw a physical and logical diagram using the equipment listed above.